AD-A203 080

AFGL-TR-88-0080(I)

Spacecraft/Environment Interactions CAE Tool User's Manual - Volume I

J. R. Lilley
S. G. Waisman
D. Peterka
D. Daou
M. J. Mandell

S-CUBED
P. O. Box 1620
La Jolla, CA 92038-1620

January 1988

Final Report
August 1986-January 1988

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

AIR FORCE GEOPHYSICS LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATED AIR FORCE
HANSCOM AIR FORCE BASE
MASSACHUSETTS 01731-5000

"This technical report has been reviewed and is approved for publication"

(Signature)
MICHAEL J. GIGER, 1Lt, USAF
Contract Manager

(Signature)
JOHN A. GAUDET, LT COL, USAF
Branch Chief

FOR THE COMMANDER

(Signature)
RITA C. SAGALYN
Division Director

This report has been reviewed by the ESD Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

Qualified requestors may obtain additional copies from the Defense Technical Information Center. All others should apply to the National Technical Information Service.

If your address has changed, or if you no longer with to be removed from the mailing list, of if the addressee is no longer employed by your organization, please notify AFGL/DAAC, Hanscom AFB, MA 01731. This will assist us in mailing a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release: distribution unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) SSS-R-87-8804	5 MONITORING ORGANIZATION REPORT NUMBER(S) AFGL-TR-88-0080(I)
6a NAME OF PERFORMING ORGANIZATION S-CUBED, A Division of Maxwell Laboratories, Inc. 6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION Air Force Geophysics Laboratory
6c. ADDRESS (City, State, and ZIP Code) P. O. Box 1620 La Jolla, CA 92038-1620	7b ADDRESS (City, State, and ZIP Code) Hanscom Air Force Base, MA 01731-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-85-C-0193
8c. ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO 63410F; PROJECT NO 2821; TASK NO 01; WORK UNIT ACCESSION NO AC

11 TITLE (Include Security Classification)
Spacecraft/Environment Interactions CAE Tool. User's Manual - Volume I

12 PERSONAL AUTHOR(S)
J. R. Lilley, S. G. Waisman, D. Peterka, D. Daou, M. J. Mandell

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Final Report	FROM Aug 86 TO Jan 88	1988 January	168

16 SUPPLEMENTARY NOTATION
Computer aided engineering

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Spacecraft charging analysis codes, NASCAP, POLAR, CAE tools

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This ~~CAE~~ tool package will aid spacecraft developers by adding a user-friendly interface to two spacecraft charging analysis codes, namely NASCAP/GEO (NASA Charging Analyzer Program, Geosynchronous Orbits) and POLAR 1.1 (Potentials of Large Orbiting Spacecraft in the Auroral Region). The software package contains four major, independent programs. They are a model definition program with a specialized interface to ANVIL 5000, separate interactive control programs for analyzing models in different environments using either NASCAP/GEO or POLAR 1.1, and a graphics display program to present the calculation results using MOVIE.BYU (DYNA-MOVIE). (KR)

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL Michael J. Giger, Lt., USAF	22b TELEPHONE (Include Area Code) (617) 377-3991 / 22c OFFICE SYMBOL AFGL/PHE

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted.
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

1. OVERVIEW OF THE CAE TOOL PACKAGE

The CAE Tool package will aid spacecraft developers by adding a user-friendly interface to two spacecraft charging analysis codes, namely NASCAP/GEO (NASA Charging Analyzer Program, Geosynchronous Orbits) and POLAR 1.1 (Potentials of Large Orbiting Spacecraft in the Auroral Region). The software package contains four major, independent programs. They are a model definition program with a specialized interface to ANVIL 5000, separate interactive control programs for analyzing models in different environments using either NASCAP/GEO or POLAR 1.1, and a graphics display program to present the calculation results using MOVIE.BYU (DYNA-MOVIE).

1.1 INTRODUCTION TO THE USER'S MANUAL

This document is intended to explain the steps necessary to use the CAE Tools to analyze the behavior of satellite models in space environments. Where possible, examples and hints have been included to make use of Tool software as easy as possible. Methods for efficiently using the analysis codes, NASCAP/GEO and POLAR 1.1 are presented, as well as information concerning the interpretation of the results from analysis codes. The manual describes the common uses, concerns, and problems associated with the analysis tools.

The typical user is presumed to have a basic understanding of the physical processes involved in spacecraft charging. It is strongly suggested that the user locate the manuals for NASCAP/GEO and POLAR 1.1 (the NASCAP

Programmers Reference Manual[1] and the POLAR Users Manual[2]). These references provide a much more detailed discussion of the phenomena, the analysis codes, and their use than is possible here. For a complete description of ANVIL 5000, the user should refer to the ANVIL 5000 manuals. For a complete description of the output display programs, the user should consult the MOVIE.BYU training text.

It is also recommended that the user be comfortable with the computing environment within which the CAE Tools have been installed. Specific operating system features are discussed only where pertinent. Please refer to the appropriate system reference manual for any operating system questions.

For questions concerning complicated or innovative uses of the analysis codes to model satellites in other than ambient environments, please see the user manual appropriate to the analysis code (the NASCAP Programmers Reference Manual or the POLAR Users Manual). Some references about the physical phenomena which are involved with spacecraft charging have been included in Appendix A.

The manual is organized in much the same way as the software. The chapters are built around the different phases a person must go through to define a satellite model, analyze the behavior of the model in an environment, and display the results.

1.2 MODEL DEFINITION

The model definition is performed using ANVIL 5000 and an optional tablet interface. This interface has been

developed using a Tektronix 4207. Other terminal types may be used provided they are supported by ANVIL 5000. The GRAPL routines which are used to define NASCAP/GEO and POLAR 1.1 recognized building blocks are available without using the tablet interface (see Chapter 4, Model Building).

Before model definition is begun, it is recommended that the user determine which of the analysis tools will be used. Since NASCAP/GEO and POLAR 1.1 have slightly different sets of recognized building blocks, some care should be taken to identify the desired analysis program in advance. If old Object Definition Files are available, they may be used directly by GEOCAT or POLCAT, the analysis tool control programs.

If a satellite is to be designed from scratch, several factors need to be considered. Objects are defined within a 17 x 17 x 33 cubical mesh. The scale length of the model, the size of the mesh in meters, should be chosen so that the spacecraft will fit within this grid. The scale length should be chosen to allow a fairly accurate representation of the geometry of the satellite. Surface materials and their locations also influence the charging behavior of the spacecraft. Finally, the layout of the underlying conductors needs to be known at the time of model definition.

1.3 ANALYSIS TOOL CONTROL PROGRAMS

The analysis tool control programs, GEOCAT and POLCAT, are set up to appear and behave the same. The only differences are those related to the respective analysis

codes. Differences can be expected in the recognized keywords, the analysis modules names and functions, and the contents of the data files after a calculation.

The control programs have been designed to run on alphanumeric terminals. They require, on VAX/VMS computer systems, the Screen Management Guideline which is part of the VMS 4.3 operating system. A VT100, or equivalent terminal is assumed, but any terminal supported by the operating system's screen manager should also work.

The typical sequence of events followed when analyzing an object using the analysis codes is to define the satellite, perform calculations, and study the results of the calculation. More calculations or object modifications may then be required.

Before a calculation can be performed, the satellite environment needs to be defined. An environment description includes plasma densities and energy distributions, sun location, magnetic fields, and satellite velocity. The operating voltage of the spacecraft's conductors must also be set. Finally, the calculation parameters, such as timestep and amount of desired convergence, need to be chosen. Each of the analysis codes have their own parameters and the appropriate user's manual should be consulted.

The control programs are used to translate the IGES file generated by ANVIL 5000 into a building block definition of the satellite. The surface materials are edited into the object definition file. Material properties are taken from the material database files and also placed in the object definition.

A "zero-dimensional" analysis of surface material interaction with several plasma environments can be done using the surface charging spreadsheet (SuChgr). The floating potentials (the voltages where incident currents balance secondary currents) can be used as guidelines for planning and interpreting the three-dimensional analysis.

After defining an object and an environment, the analysis codes perform full three-dimensional calculations of the spacecraft's interactions with the plasma environment. The control programs provide a simple rundeck editor and job controller for running a batch analysis calculation. NASCAP/GEO and POLAR 1.1 are keyword input driven. The rundeck editor can check the keywords for validity. When the input rundeck is ready, the job control module will set up and start the analysis calculation. The job control module can also smoothly stop the run if so desired. Run status information is also provided.

Postprocessing tools available from within the control programs are an output file pager and the surface data viewer, TERMTALK, for NASCAP/GEO calculations.

Additional modules within the tools provide the means to maintain files and directories and to issue system functions. A form is also available which can be used to tailor tool environment features such as the directory search order, the location of the home directory, and the interaction mode with the user.

Context sensitive help messages are available throughout the control programs.

1.3.1 GEOCAT (NASCAP/GEO)

GEOCAT is the control program used to work with NASCAP/GEO analysis code. The surface charging module, SuChgr, uses environment definitions recognized by NASCAP/GEO. Five modules can be run from GEOCAT: IGES translator, NASCAP, CONBYU (creates BYU format potential contour plots), BYUPOT (creates BYU format surface potential plots), and BYUMAT (creates BYU format material plots). The TERMTALK postprocessor is also available as a postprocessor within the control program.

The purpose of NASCAP/GEO is to perform a dynamic, fully three-dimensional simulation of electrostatic charging processes for an object in space (magnetospheric) or ground test (electron beam) environments. In particular, NASCAP/GEO predicts surface potentials on spacecraft or test objects, identifies possible discharge sites, predicts satellite response to environment changes, and predicts and interprets particle detector results.

1.3.2 POLCAT (POLAR 1.1)

POLCAT is the control program used with the POLAR 1.1 analysis program. The surface charging module, SuChgr, uses environments found in low earth, polar orbits. The four POLAR 1.1 modules, VEHICL, ORIENT, NTERAK and SHONTL are controlled from within POLCAT.

POLAR 1.1 is designed to model spacecraft/environmental interactions in the auroral regions. It models flowing plasmas with magnetic fields and high energy electron spectra. Although it is best suited for space

charge limited collection problems, orbit limited cases can also be studied. See Section 3.2 or the POLAR User's Manual for further discussions of the applications and algorithms of POLAR 1.1

1.4 DISPLAY OF CALCULATION RESULTS

No assumptions have been made about the graphics device to be used by the display files created by GEOCAT and the analysis code POLAR 1.1. Any graphics hardware environment which supports MOVIE.BYU, DYNA-MOVIE, or any other display program which accepts the BYU file format may be used.

The types of output available from the control programs and analysis codes include two-dimensional potential contour and density plots, and three-dimensional surface potential, material, and current plots. MOVIE.BYU and DYNA-MOVIE can be used to display all of these calculation results.

For NASCAP/GEO, older, stand-alone graphics programs provide additional object-display and potential-contour capability. Interfaces to various graphics devices are provided. These interfaces also serve to display graphics created within NASCAP/GEO.

2. NASCAP/GEO ANALYSIS TOOL OVERVIEW

Several steps are involved in a typical NASCAP/GEO calculation. The first step is defining a valid NASCAP/GEO model or object definition. This is accomplished graphically using the object manipulation and display capabilities of ANVIL 5000 and a custom graphics tablet.

After the object has been defined, the NASCAP/GEO Control Program, GEOCAT, is used to generate rundecks for calculation runs, to start, halt, and monitor the NASCAP/GEO batch jobs, and to interact with the post processing program to generate text and graphic output files.

Finally, the graphics output files are viewed using MOVIE.BYU or DYNA-MOVIE.

2.1 WHAT IS NASCAP?

NASCAP, the NASA Charging Analyzer Program, is a computer program designed to simulate spacecraft charging. Spacecraft charging is the build-up of electrostatic potentials on the surfaces of spacecraft exposed to a plasma environment. This occurs when charged particles from the plasma collect on the exposed surface. Both the sign and the magnitude of the potential acquired from exposure to the same plasma may differ for different surface materials, or for different areas of the same material due to shadowing or electrostatic effects. Thus a complicated object composed of more than one material may charge non-uniformly leading to differential charging, i.e., potential differences between different parts of the object. Differential charging can cause electrical discharges that may be damaging to satellite systems.

For objects as structurally complicated as man-made satellites, predicting their interaction with a surrounding plasma in a test tank or space environment becomes a very complex problem. The purpose of NASCAP is to solve this problem and calculate such observable quantities as electric potentials and currents to and from the spacecraft. NASCAP is an important tool for the analysis of spacecraft charging and the interplay between the various mechanisms responsible.

2.2 THE PHYSICS OF SPACECRAFT CHARGING

The atmosphere around the earth at geosynchronous altitude consists of a low density, energetic plasma. Both electron and ion components of the plasma have similar Maxwellian-like spectra, so that the flux of the much lighter electrons greatly exceeds that of the ions. If the collection of charge were due only to primary plasma currents, all materials would charge to negative potentials of a few times the plasma temperature. However, the impact of both primary electrons and ions on the exposed surface causes the ejection of low energy (<10 eV) secondary electrons into space. Impacting electrons can also be reflected as backscatter. These mechanisms all act as additional sources of positive current. In sunlight, photoelectrons ejected from the surface also act as a source of positive current. Photoelectrons, like secondary electrons, have low energy. Finally, current may flow to and from a surface from other parts of the object via bulk and/or surface conduction. The net current (i) to any surface is the algebraic sum of these contributions:

$$i_{net} = i^{electrons}_{primary} + i^{ions}_{primary} + i^{electrons}_{secondary} + i^{ions}_{secondary}$$

$$+ i^{electrons}_{backscatter} + i_{conductivity} + i_{photoemission}$$

If i_{net} is initially negative, the exposed surface will begin to acquire a negative potential. As the magnitude of the potential increases the net current is attenuated, until it eventually approaches zero and the surface potential remains at a steady equilibrium value. Equilibrium potentials exceeding -10 kV have been observed in geosynchronous earth orbit.

If i_{net} is initially positive, the exposed surface will begin to acquire a positive potential. However, large positive equilibrium potentials are not normally achieved. This is because low energy secondary and photoemissions provide the dominant contribution to a positive current. As soon as the surface reaches a potential greater than the energy of the emitted electrons (5 or 10 eV) they can no longer escape and charging stops. In this case equilibrium is determined by the suppression of low energy emission due to the surface's own electric field. A similar suppression effect may occur due to the electric fields of neighboring negatively charged surfaces. This adds to the complexity of the situation for charging of complicated objects and makes spacecraft charging a truly three-dimensional problem.

2.3 NASCAP CAPABILITIES

NASCAP is a collection of the various models and algorithms needed to simulate the charging of a complex object. The various formulations are written to levels of accuracy and approximation appropriate to solving problems for geosynchronous-like conditions in a reasonable amount of computer time. The NASCAP user has a great deal of flexibility in applying these capabilities to his particular problem. Among NASCAP's capabilities are:

- To define complex objects from fairly simple input.
- To define properties of materials relevant to spacecraft charging.
- To calculate electrostatic potentials around complex objects.
- To calculate shadowing of one part of an object by another.
- To calculate primary currents incident on spacecraft surfaces from a plasma or from point sources.
- To calculate secondary and backscattered electron currents.
- To calculate conductivity, biasing, and grounding currents.
- To calculate charge accumulation and resulting surface potentials.
- To calculate trajectories of charged particles incident upon, or emitted from, specified surfaces.
- To meaningfully communicate results through printed output, graphical output, and interactive post-processors.

These capabilities satisfy the requirements for study of the processes and the consequences of spacecraft charging.

2.4 THE NASCAP PHYSICAL MODEL

NASCAP objects are defined within a three-dimensional cuboidal grid (rather like a shoe box). The grid is composed of many thousands of identical cubes or volume elements stacked together. Objects are defined by filling or partially filling the cubes. For example, a "quasi-sphere" is shown in Figure 2.1. The exterior surface of the filled volume elements form the exposed surface of the object. Thus, the object surface consists of rectangular or triangular patches called surface cells. In addition to the cubic elements, NASCAP allows arbitrarily thin cylindrical booms and thin plates to be defined. The definition of NASCAP objects is discussed in detail in Section 4.

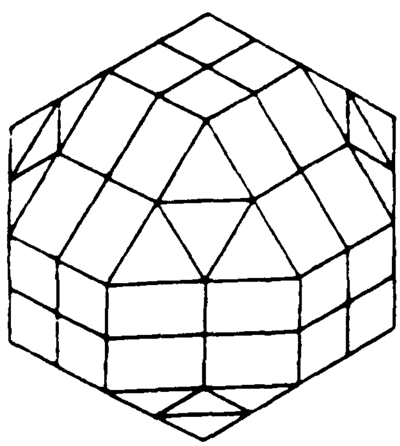

Figure 2.1. A quasi-sphere.

NASCAP calculates the potentials and currents for an object that has been exposed to a plasma environment for a chosen period of time or timestep. The initial conditions at the beginning of the timestep may be specified by the

user or may be remembered from the previous timestep calculation. Similarly the results predicted for the end of the current timestep may be used as the initial condition for the next, and so on. By using a sequence of timesteps, a user may follow the dynamics of the approach to equilibrium as well as being able to examine the equilibrium state itself. The shorter the timesteps chosen, the more will be needed to reach equilibrium and the greater the detail of the dynamic charging behavior calculated.

For each timestep, or cycle, NASCAP calculates the total amount of charge that collects on each surface cell. This is determined from the net current at the beginning of the cycle, taking into account all of the contributions mentioned above. The variation of the net incident current as the surface potential changes during the cycle can also be taken into account. The charge collected is translated into a new set of surface potentials via a detailed resistive-capacitive electrical model of the satellite. Poisson's equation is then solved using the new (fixed) surface potentials to give new updated potentials in the space surrounding the object. The new potential and electric field values imply a new set of currents in the next cycle. Equilibrium is achieved when currents and potentials reach steady values for consecutive timesteps. The details of the many sophisticated physical models that are part of NASCAP are discussed in the later chapters. However, there are a number of assumptions built in that define the physical regime where NASCAP works best.

2.5 THE NASCAP PHYSICAL REGIME

NASCAP assumes orbit-limited spherical probe current collection. This is a good approximation for convex objects with radius of curvature smaller than the Debye length of the ambient plasma. Hence NASCAP works well for small objects (with dimensions of a few meters or less) in geosynchronous orbit (where Debye lengths are typically hundreds of meters). While NASCAP can simulate charging, taking into account a space charge sheath surrounding the object, it is primarily designed for the low density, high temperature plasmas found at geosynchronous altitudes where space charge can be ignored. The range of physical regimes where NASCAP can be most profitably used is discussed at length in Reference 8.

2.6 GRAPHICS OUTPUT

NASCAP/GEO provides electrostatic potential contour plots, surface potential plots and surface material plots. For more information, please refer to Section 6 and the NASCAP Programmers Reference Manual.

3. POLAR 1.1 ANALYSIS TOOL OVERVIEW

Several steps are involved in a typical POLAR 1.1 calculation. The first step is defining a valid POLAR 1.1 model or object definition. This is accomplished graphically using the object manipulation and display capabilities of ANVIL 5000 and a custom graphics tablet.

After the object has been defined, the POLAR 1.1 Control Program, POLCAT, is used to generate rundecks for calculation runs, to start, halt, and monitor the POLAR 1.1 batch jobs, and to interact with the post processing program to generate text and graphic output files.

Finally, the graphics output files are reviewed using MOVIE.BYU or DYNA-MOVIE. The following is a brief overview of the capabilities and models of POLAR. For further information, please see the POLAR User's Manual.

3.1 MODEL DEFINITION

POLAR 1.1 defines models within a three-dimensional cubic grid. Objects are comprised of filled elements, partially filled elements, and their planes. Partially filled elements are those formed by passing a plane through three or four corners of an element. Thin planes are the square face of a cube or the diagonal plane formed by cutting a cube into two wedges. These elements are discussed further in Section 4 and the POLAR User's Manual.

The computational grid structure of POLAR 1.1 is composed of two regions, the object grid and the extended computational grid. POLAR extends the grid along the z-axis

in order to optimally model flowing plasmas. These grid extensions may be staggered in order to provide a means of modeling any flow direction (see Figure 3.1).

For additional discussions of POLAR 1.1 models and mesh definitions, please refer to Section 4 and the POLAR User's Manual.

3.2 MODEL ANALYSIS

The POLAR code models the behavior of objects in the ionospheric environment found in low earth polar orbits. Features which are modeled include multi-component, high energy electron spectra, flowing ion species, magnetic fields, spacecraft generated wakes, and surface charging including secondary transport along surfaces. The physics of spacecraft/environment interactions in these orbits is dominated by space charge and magnetic effects.

The POLAR code makes various assumptions which enable it to perform three-dimensional charge calculations in relatively short Debye length plasmas. In this section we examine the component physical models and discuss their validity. While each is addressed separately, the code achieves a self-consistent solution by various levels of iteration.

One major, overriding assumption should be identified before the component by component description, which is that all time dependence on the scale of particle dynamics is ignored. This means that particles see spatially dependent

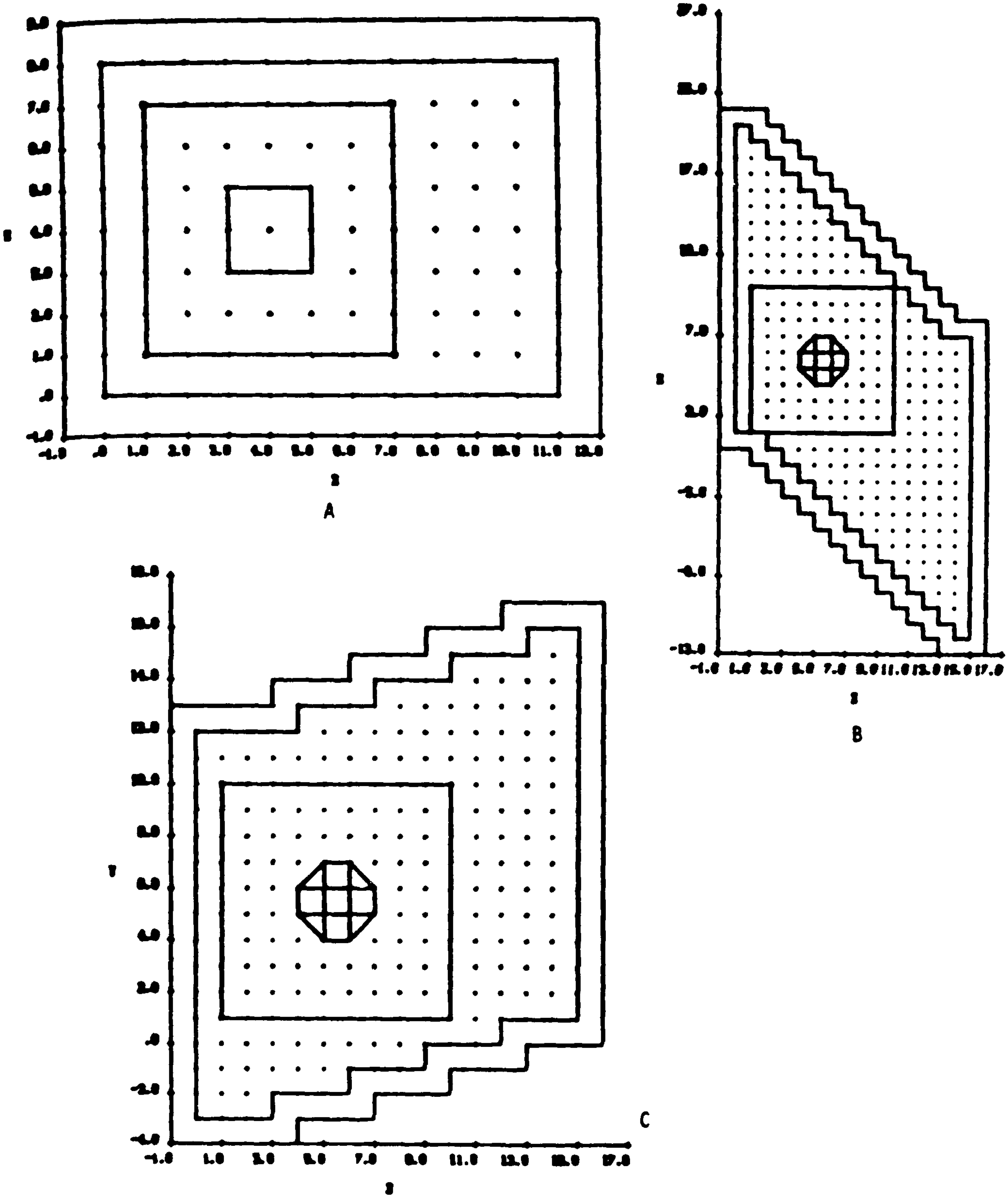

Figure 3.1. (A) Cube moving to left. (B) Quasisphere moving to left and rising. (C) Quasisphere moving to left and falling.

but time independent fields for the period they are near the orbiting vehicle. As such, all plasma oscillations, including electron and ion modes are precluded. Thus, oscillations in the wake or at leading edges will not be predicted by the POLAR code.

3.3 THE POLAR PLASMA ENVIRONMENT

POLAR can model a wide variety of plasma environments from reasonable combinations of the following populations:

<u>Ions</u>

Cool Maxwellian ions.
Cool Maxwellian protons.

Both the protons and ions are assumed to be isotropic in the plasma frame. The relative densities are controlled by inputting the density ratio with the total constrained to equal the ambient electron density. Both populations have temperatures equal to the temperature of the cool electrons, temperature 1. During wake calculations the ion temperature can be defined to be different than the electron temperature.

<u>Electrons</u>

Cool ambient Maxwellian, temperature 1, density 1.
Suprathermal, power law distribution of energies.
Hot Maxwellian, temperature 2, density 2.
Energetic, Gaussian distribution of energies.

The cool Maxwellian population is considered isotropic in the plasma frame. The other, more energetic populations may be given field-aligned and loss-cone distributions in the future but are presently considered isotropic only.

3.4 PLASMA POTENTIALS

The other major assumption is that the only fields of major importance are the static electric fields obtainable from Poisson's equation and the earth's magnetic field. The only velocity related field included is that induced by **v** x **B** on conducting surfaces. The frame of reference is chosen to be the stationary plasma, so that **v** x **B** effects appear on the vehicle as boundary conditions. The plasma at infinity is defined to be at zero potential.

Plasma potentials are obtained from Poisson's equation

$$-\nabla^2\theta = \lambda^{-2}(n_i - n_e)$$

where $\theta = eV/kT$ is the dimensionless potential, λ is the Debye length ($\lambda^2 = \epsilon_o\ kT/Ne^2$), and n_i and n_e are the appropriate ion and electron charge densities. Contributions from hot auroral electrons and particles backscattered from the vehicle are neglected except for electron secondaries generated during electron collection. Poisson's equation is solved using either fixed potential or fixed normal electric field boundary conditions on a surface by surface basis, as appropriate.

3.5 GRAPHICS OUTPUT

The SHONTL module of POLAR provides a means for plotting two-dimensional contour plots of the three-dimensional electrostatic potential and normalized charge density data. New modifications permit plotting of any of the surface data stored in the POLAR database. Please refer to the POLAR User's Manual for additional information on SHONTL and available surface data.

4. MODEL BUILDING

INTRODUCTION

This chapter is designed to help prevent some of the possible confusion which may arise when the user first starts using the ANVIL 5000 Graphics Interface. It should function both as a reference for specific ANVIL 5000 functions and all the GRAPL programs, and as an aid to beginning users.

Section 4.1, RECOGNIZED NASCAP/GEO AND POLAR 1.1 MODEL BUILDING BLOCKS, introduces the idea of using building blocks to define objects, describes them, and concludes with

Section 4.2, an INTRODUCTION TO ANVIL 5000 AND GRAPL, talks about the amount of background with ANVIL 5000 the user should have; the software and hardware setup necessary to run the Graphics Interface; and how to invoke the Interface depending on the user's setup.

Section 4.3, USING THE ANVIL 5000 GRAPHICS INTERFACE, contains important background information that the user will need to know before he can successfully define objects with ANVIL 5000. This is followed by an extensive introduction to the key terms used in ANVIL 5000, and some of the basic commands which will be used time and time again. Finally, we include a word about Tablet versus No Tablet, and include a table of tablet choices and their equivalent menu keystrokes.

SECTION 4.4, GRAPL programs and building blocks, will probably be the most used section in this chapter. It contains important information for all GRAPL programs. This should make the user's life a lot easier by anticipating some of the difficulties that may arise. The bulk of this section is an exhaustive discussion of each GRAPL program, with a Trouble-shooting Modeling Error section and suggestions at the end of each.

Section 4.5, ADVANCED HINTS AND TRICKS, closes with some useful features of the ANVIL 5000, as well as references to other sources of information.

HOW TO USE CHAPTER 4.0 MODEL BUILDING

Read Ahead

We recommend that most of Sections 4.1, 4.2 and 4.3, and the beginning of Section 4.4 be read through once as background, before the user gets on the terminal.

Then, after the user begins defining building blocks, it will probably be helpful to go back to specific sections. Specifically, the discussions at the beginning of Section 4.4 will be better understood after a user has already tried running one or more GRAPL programs.

Don't Panic

The ANVIL 5000 Graphics Interface is designed to be user oriented. This means that ANVIL 5000 will prompt the user for responses at each step. For most situations, once the user learns a few of the terms used in the Graphics

Interface, it will be possible to figure out what to do and what is going on by following the instructions on the screen.

4.1 RECOGNIZED NASCAP/GEO AND POLAR 1.1 MODEL BUILDING BLOCKS

The first step in order to use NASCAP/GEO or POLAR 1.1 is to define a valid object definition (objdef) file. The user will be able to do this graphically using the ANVIL 5000 Graphics Interface. The ANVIL 5000 Part File is then written as an IGES file using the Interface. Then the appropriate CAE Tool can be executed to convert the IGES file to an objdef file.

This section introduces the building blocks generated when defining objects which will be run through NASCAP/GEO or POLAR 1.1. This is followed by exceptions to both NASCAP/GEO and POLAR 1.1. Finally, we include a list of limitations which the user should be aware of in object definitions.

Most of the information in this section is a summary of the chapters dealing with this same topic in the NASCAP/GEO and POLAR 1.1 manuals. It is a general description of the way in which objects are defined. If more information is needed, the user should refer to the "NASCAP PROGRAMMER'S REFERENCE MANUAL," Chapter 3, or to the "POLAR USER'S MANUAL," Chapter 6.

For a detailed description of how each building block is actually defined using the ANVIL 5000 Graphics Interface, see Section 4.4.

4.1.1 Defining Objects: Building Block History

All NASCAP/GEO and POLAR 1.1 objects are confined to the space inside the innermost 17 x 17 x 33 grid. This is the grid in which the user will define objects in ANVIL 5000. Only BOOMS (see Section 4.1.3) are allowed to extend into outer grids. If "empty space" and "object" both coexist in the same computational space what makes objects distinguishable? The answer is that NASCAP/GEO and POLAR 1.1 can distinguish between volume elements that are <u>filled</u> (with object) and those that are <u>empty</u> (except for ambient plasma, of course). Once we have this distinction it is easy to see how objects can be constructed by filling in collections of volume elements. For example, a simple cuboid may be constructed by filling in 2 x 3 x 4 = 24 elements as shown in Figure 4.1.1.

While arrangements of completely filled and completely empty cubes can be quite versatile in representing objects of many different shapes, more sophisticated representations are possible if we allow cubes to be partially filled (or, as a pessimist might say, partially empty). Only three partially filled cubes are allowed. These are shown in Figure 4.1.2.

While it is easy to see how objects might be constructed by filling or partially filling individual volume elements, a command structure that required the user to specify every element comprising an object would be very cumbersome to use.

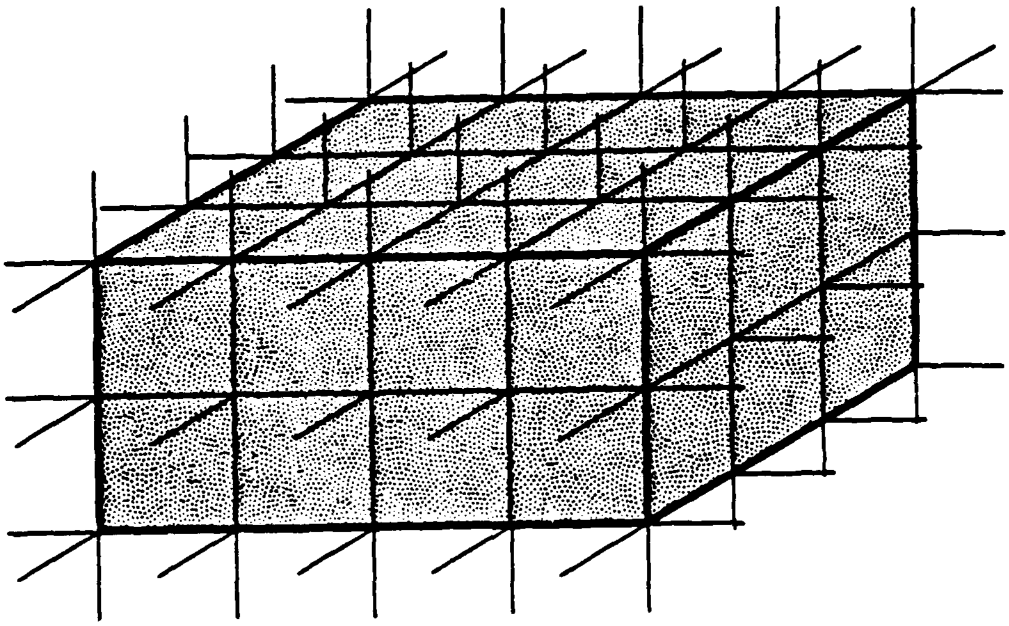

Figure 4.1.1. Cuboid made by filling in twenty-four volume elements.

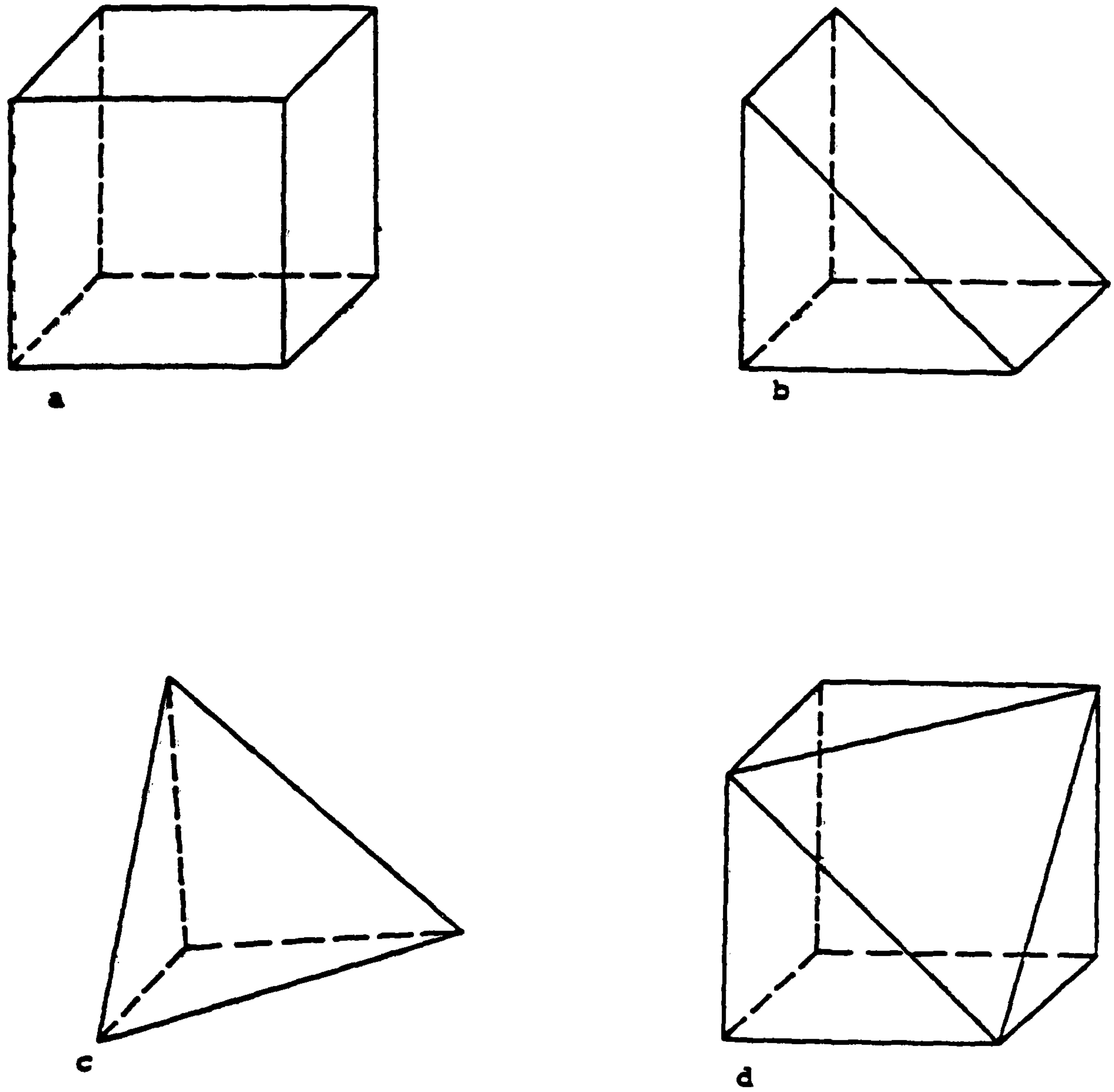

Figure 4.1.2. Four shapes of volume cells considered by the NASCAP/GEO and Polar 1.1 code: (a) empty cube; (b) wedge-shaped cell with 110 surface; (c) tetrahedron with 111 surface; (d) truncated cube with 111 surface.

4.1.2 Building Blocks

To greatly simplify the user definition of objects NASCAP/GEO and POLAR 1.1 pre-define commonly used shapes built up from individual elements. These shapes are called NASCAP/GEO or POLAR 1.1 BUILDING BLOCKS. NASCAP/GEO building blocks are:

Rectangular Parallelepiped
Octagon
Quasisphere
Tetrahedron
Wedge
FIL111
Flat Plate
Boom
Transparent Antenna
    (Aplate, Aslant, Atet)

There are eight in POLAR 1.1:

Flat Plate
Slanted Plate
Rectangular Parallelepiped
Octagon
Quasisphere
Tetrahedron
Wedge
FIL111

These are shown in Figure 4.1.3 (a) and (b). These basic shapes can be defined to be any size (within the inner grid). NASCAP/GEO automatically includes the correct number of individual elements for the size of building block chosen by the user.

These kinds of building blocks can be easily defined with the ANVIL 5000 Graphics Interface.

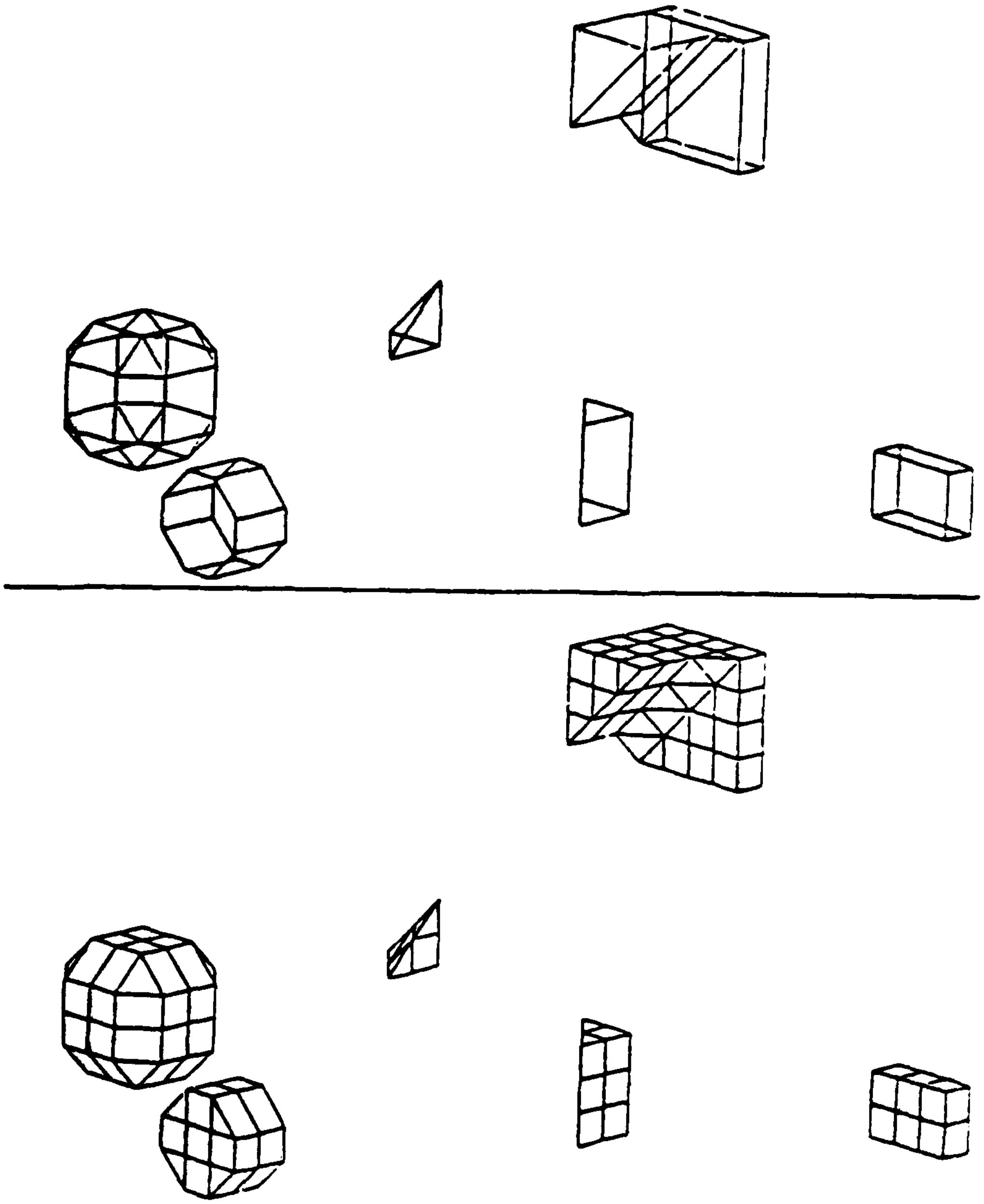

Figure 4.1.3a. The six building block types used in NASCAP/GEO are shown here. The uppermost object shows a FIL111 smoothing a corner. Below, from left to right are quasi-sphere, octagon right cylinder. telrahedron, wedge, and rectangular parallelepiped.

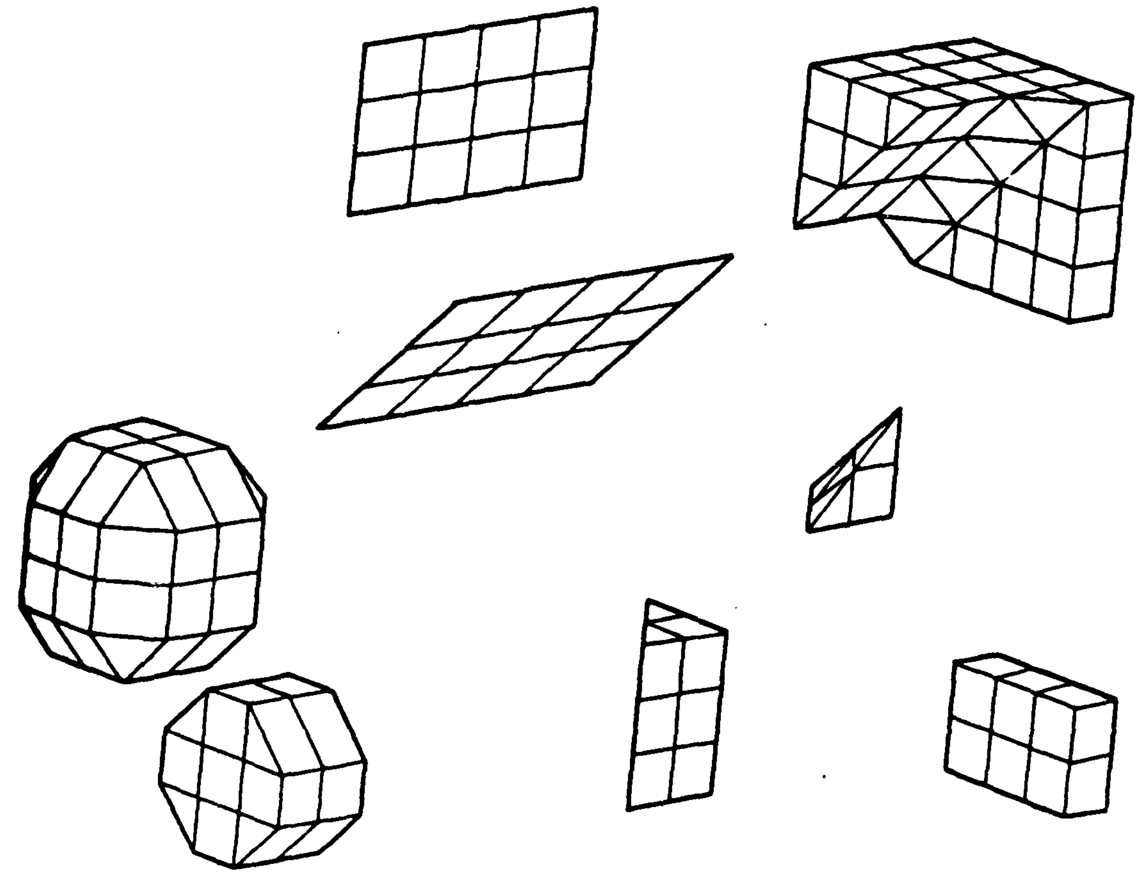

Figure 4.1.3b. The eight building block types used in POLAR 1.1 are shown here. The uppermost object shows a FIL111 smoothing a corner. Below, from left to right are quasi-sphere, octagon right cylinder, tetrahedron, wedge, and rectangular parallelepiped.

TABLE 4.1.1 (a)

NASCAP/GEO BUILDING BLOCKS AND THEIR GRAPL PROGRAMS

Keyword	Building Block Description
BOOM	Long thin BOOM.
FIL111	Smooth inside of a diagonal corner.
OCTAGON	Right octagonal cylinder.
PATCHR	Surface of a rectangle.
PATCHW	Diagonal face of a wedge.
PLATE	Arbitrarily thin plate or cuboid.
QSPHERE	Quasisphere.
RECTAN	Cuboid or rectangular parallelepiped.
TETRAH	Tetrahedron.
WEDGE	Wedge derived from half a cube.
APLATE	Transparent antenna plate.
ASLANT	Transparent antenna slanted plate.
ATET	Transparent antenna tetrahedral plate.

TABLE 4.1.1 (b)

POLAR 1.1 BUILDING BLOCKS AND THEIR GRAPL PROGRAMS

Keyword	Building Block Description
FIL111	Smooth inside of a diagonal corner.
OCTAGON	Right octagonal cylinder.
PATCHR	Surface of a rectangle.
PATCHW	Diagonal face of a wedge.
PLATE	Arbitrarily thin plate or cuboid.
QSPHERE	Quasisphere.
RECTAN	Cuboid or rectangular parallelepiped.
SLANT	Thin plate slanted at 45°.
TETRAH	Tetrahedron.
WEDGE	Wedge derived from half a cube.

4.1.3 BOOMS, PLATES AND PATCHES

A careful inspection of Table 4.1.1 (a) and (b) will show that there are some building blocks that are not derived from cubic volume elements. These are the BOOM, PLATE, PATCHR and PATCHW for NASCAP/GEO, and PLATE, SLANT, PATCHR and PATCHW for POLAR 1.1.

BOOMs are long cylindrical projections that may have any radius less than one grid unit. They may only lie along the X, Y, or Z directions. Unlike any of the other building blocks they may extend beyond the innermost grid.

PLATEs are arbitrarily thin cuboids (RECTANs). They are assumed to have only a top and a bottom, the sides being of negligible height. Flat plates always lie in one of the axis planes (XY, XZ, YZ). SLANTed plates lie along one axis, and at a 45° angle to the other two.

PATCHR and PATCHW are the surfaces only of a cuboid and wedge, respectively. They are used to change the surface material patterns of existing building blocks and should never be defined in spaces not already occupied by solid objects.

4.1.4 FIL111 and Transparent Antenna Surfaces

4.1.4 (a) FIL111

FIL111 is a special shape designed to fill in "steps" whose corner line runs at 45° to the grid lines in any axis plane (i.e., XY, ZY, XZ) (Figure 4.1.4 (a)). There are two kind of "steps" that can occur between NASCAP/GEO and POLAR 1.1 building blocks. For example, a small cuboid on top creates four "steps" that lie along grid lines (Figure 4.1.4 (b). These may be "filled in" or smoothed by defining a WEDGE to lie along the corner line of the step. A second type of step is possible however when, for example, a wedge or octagon is defined to sit on top of another building block. These steps have corner lines that run at 45° between grid lines. This is shown in Figure 4.1.4 (c).

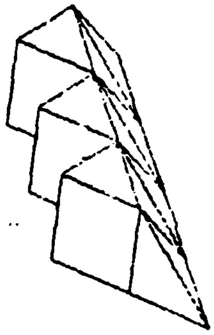

Figure 4.1.4a. A FIL111 building block all by itself.

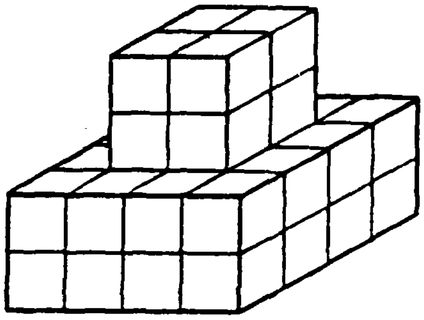

Figure 4.1.4b. "Steps" along grid lines.

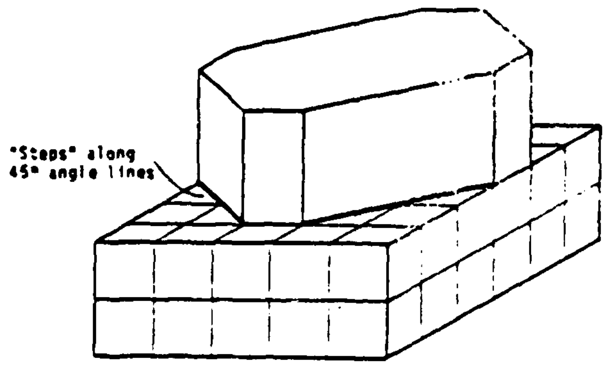

Figure 4.1.4c. "Steps: along 45° angle lines.

Such steps can be smoothed or filled in by a combination of tetrahedra and truncated cubes. This combination is supplied as the building block FIL111.

4.1.4 (b) Transparent Antenna Surfaces (NASCAP/GEO only)

Antenna surface cells may be square (defined by the APLATE subroutine), rectangular (defined by the ASLANT subroutine), or equilateral triangle (defined by the ATET subroutine). No provision is made for right triangle antenna cells. Antenna surface cells are automatically treated as two-sided by NASCAP/GEO; only one side of the surface should be defined. Antenna surfaces should not be used to supersede solid surfaces, although solid surfaces may supersede antenna surfaces. HIDCEL draws the cell outlines of antenna cells, except, of course, where they are shadowed by solid objects. For line-plot devices such drawings can be a bit messy; plots are far better on color-fill devices. For material plots, mesh surfaces are treated as nontransparent.

4.1.5 NASCAP/GEO Exceptions

In NASCAP/GEO, the SLANT building block is not allowed.

NASCAP/GEO does have several building blocks in addition to those allowed by POLAR 1.1. The new ones are the BOOM and three kinds of antennas - ATET, ASLANT and APLATE.

4.1.6 POLAR 1.1 Exceptions

POLAR 1.1 does not allow the following building blocks:

BOOM
ATET
APLATE
ASLANT

Also, the user may not define a thin triangle in POLAR 1.1. (Thin triangles can be defined using the GRAPL program WEDGE. See Section 4.4.5)

4.1.7 Limitations in Object Definition

Any and all illegal objects defined with ANVIL 5000 will be caught by the CAE Tool which writes the objdef file. However, the user can save a lot of time and effort if certain restrictions intrinsic to NASCAP/GEO and POLAR 1.1 are known before the graphics definition procedure.

The first list (Numbers 1 - 4, see below) includes limitations applicable to both NASCAP/GEO and POLAR 1.1. The second list (Number 5) includes limitations which only apply to POLAR 1.1. The third list (Numbers 6 - 10) includes limitations which only apply to NASCAP/GEO, and deals primarily with BOOM related rules.

It is probably fair to say that you can link building blocks together and nine times out of ten there will not be a problem. This section deals with the other one time out of ten, when what appears to be a perfectly reasonable

combination of building blocks is rejected by OBJDEF. We itemize here a rather formidable list of object definition "don'ts." However, you should remember that it takes hard work to break more than one or two of these rules defining any one object if you use a little common sense.

List of Limitations in NASCAP/GEO and POLAR 1.1

1. All exposed surfaces must be assigned materials.

2. Thin plates sharing the same volume element can do so only if the TOP face of one shares volume with the TOP of the other, or the BOTTOM face of one shares volume with the BOTTOM face of the other. TOP faces may not share volume elements with BOTTOM faces.

3. Thin plates may only intersect each other at the edges or corners.

4. Double points must be assigned TOP and BOTTOM sets (see Section 4.1.8).

List of Limitations only in POLAR 1.1

5. The object must not touch the object grid boundary planes at any point.

List of Limitations only in NASCAP/GEO

6. No surface may lie in the planes that form the boundary of the inner mesh. Surfaces may touch the boundary planes at a point or line.

7. Booms may not lie in the boundary planes. Booms may cross a boundary plane but only from an inner to an outer grid and not vice versa.

8. Booms may not lie along the edges of filled or partially filled volume elements or pass through objects.

9. Two booms may not share the same volume element; i.e., two parallel booms must be at least two grid units apart, and two perpendicular booms may not intersect.

10. A boom cannot share a volume element with the BOTTOM of a thin plate.

(Rules 2 through 4 and Rule 10 are all manifestations of conflicts involving double and triple points.)

4.1.8 Double Points

Thin plates may have different potentials on their two surfaces, yet they occupy only one plane of grid points. This grid points must therefore be associated with two distinct sets of potentials. For this reason they are called double points. The two sets of potentials associated with each half of the double points are distinguished by calling one set 'TOP' and one set 'BOTTOM'. Recall (4.4.7) that the surfaces of a thin plate may be defined as 'TOP' or 'BOTTOM' regardless of whether their surface normal points along a positive or negative axis direction: The TOP and BOTTOM definition refers to the (arbitrary) choice of which set of potentials (TOP or BOTTOM) to associate with each surface. When double points share a volume element they must all be of the same type; i.e., all TOP or all BOTTOM. This is the basis for rule 2 in Section 4.1.7.

Double points also occur when other building blocks touch in such a way that their single points come together to form a common vertex of two "disjoint" volume elements. By "disjoint" volume elements we mean elements physically separated from each other by solid surfaces. This is shown

for two cuboids touching along one edge only in Figure 4.1.8. The row of points along the touching edges are double points and one set must be defined as BOTTOM. This may be done by defining a thin plate touching the common edge. If the exterior surface of the plate pointing into one of the disjoint volumes is 'BOTTOM' then the half of the double point associated with the other disjoint volume becomes 'TOP'.

Because of the way surface cell potentials are assigned to the grid points the edges of thin plates are only single points. However, a thin plate touching another building block (not BOOMs!) with its edge creates a row of double points similar to that caused by two cuboids touching at an edge (Figure 4.1.9). These double points are automatically assigned TOP and BOTTOM sets.

4.1.9 Triple Points

A triple point is said to occur when a vertex is common to three or more disjoint volume elements. Triple points are illegal! The easiest way to get a triple point is to define one thin plate passing through another. This is not allowed (rule 3, Section 4.1.7).

4.1.10 Limit on Conductors and Materials

NASCAP/GEO and POLAR 1.1 allow conductor and material numbers between 1 and 15.

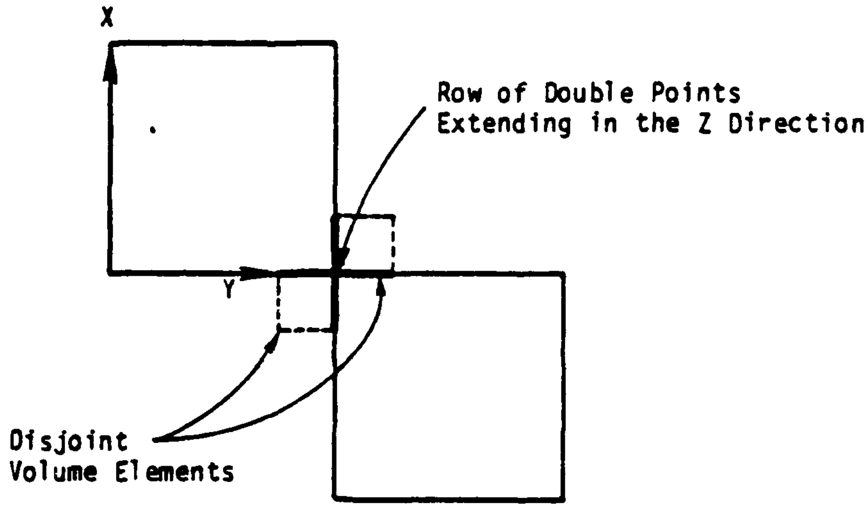

Figure 4.1.8. Profile of two cuboids sharing a common edge and resultant double points. Heavy lines show possible orientations for the definition of a thin plate to resolve the conflict.

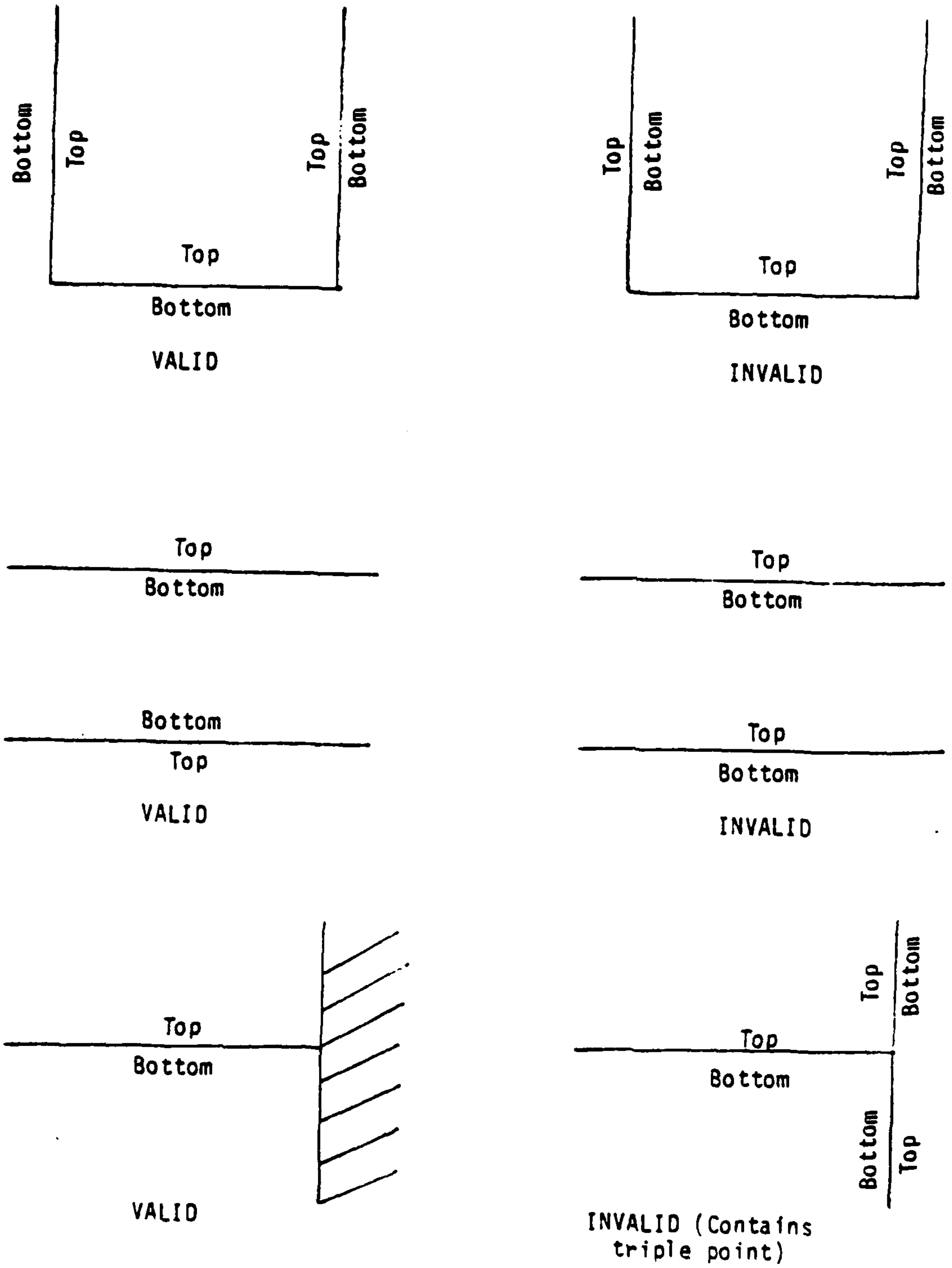

Figure 4.1.9. Examples of valid and invalid TOP/BOTTOM specifications.

4.2 AN INTRODUCTION TO ANVIL 5000 AND GRAPL

4.2.1 Running ANVIL 5000: What You Need to Know

It is apparent that the user cannot use ANVIL 5000 to define NASCAP/GEO of POLAR 1.1 objects (or any other objects) unless (a) ANVIL 5000 is resident on his system, and (b) he has the privilege to run it. Therefore, the user's first step should be to ensure ANVIL 5000 is accessible. It is also advisable to obtain the documentation for ANVIL 5000. Many points not covered here can be found there.

In constructing the ANVIL 5000 Graphics Interface, we have tried to minimize the knowledge of ANVIL 5000 required of the user. However, if the user is not familiar with ANVIL 5000, we recommend that he work through the first few lessons of the ANVIL 5000 tutorial. He should become familiar with choosing menu items, with the use of "REJECT" (left bracket, "[") to exit a menu, with the use of "END" (right bracket, "]") to end data entry, with the use of ctrl-F to return to the top menu, and with the method of indicating coordinates on the screen.

4.2.2 Preparing to run the ANVIL 5000 Graphics Interface: Software and Hardware Needed

The ANVIL 5000 Graphics Interface consists of

(1) An empty "PART" file, NASCAP.PRT, containing gridding, views, and other modals designed for ease of defining NASCAP/GEO and POLAR 1.1 objects;

(2) A set of GRAPL programs, *.GPL (and their sources, *.GSR) for defining the various NASCAP/GEO blocks; and

(3) A tablet overlay file, OBJDEF.TBN, which enables the user to shortcut the commonly used menu paths.

The user should have the following files in the directory from which he intends to run ANVIL 5000:

ANVIL5K.CFG	APLATE.GPL	APLATE.GSR	ASLANT.GPL
ASLANT.GSR	ATET.GPL	ATET.GSR	BLNKLS.GPL
BLNKLS.GSR	BOOM.GPL	BOOM.GSR	CNDCOL.GPC
CNDCOL.GSR	COLORD.GPL	COLORD.GSR	DELBLK.GPL
DELBLK.GSR	FIL111.GPL	FIL111.GSR	GRID.GPL
GRID.GSR	ICOLINTS1.GPL	INITBS.GPL	INITBS.GSR
NASCAP.PRT	OCTAGO.GPL	OCTAGO.GSR	PATCHR.GPL
PATCHR.GSR	PATCHW.GPL	PATCHW.GSR	PLATE.GPL
PLATE.GSR	PPT1T8.GPL	PPT1T8.GSR	QSPHER.GPL
QSPHER.GSR	RECTAN.GPL	RECTAN.GSR	SLANT.GPL
SLANT.GSR	TETRAH.GPL	TETRAH.GSR	TILT1.AVW
TILT2.AVW	TILT3.AVW	WEDGE.GPL	WEDGE.GSR

Total of 48 files.

The user must have "RWE" privileges on all these files except NASCAP.PRT, which should be "RE" in order to avoid inadvertently writing a non-empty part into it.

In order to activate the customized tablet overlay, the variable MCS$CFG should be set to the full file name

(include directory, device, etc. names) of the ANVIL5K.CFG file. This file configures ANVIL 5000 at run time and, among other things, defines which file describes the tablet interface (OBJDEF.TBN).

Invoking the ANVIL 5000 Graphics Interface at S-CUBED

At S-CUBED we normally run ANVIL 5000 on a Tektronix 4207 terminal with a 4957 graphics tablet. Occasionally, we run it on a Tektronix 4014 without a tablet.

To gain access to ANVIL 5000, please consult your system administrator. Typically, the command "ANVIL5K" will execute the ANVIL 5000 program. After issuing the command "ANVIL5K" the user will be queried as to his hardware setup. Enter "2" for terminal type if you are using a Tektronix 4208 or 4129. If the NASCAP/GEO Tablet Overlay is to be used, the user should be sure that it is firmly and correctly affixed to the graphics tablet.

While waiting for ANVIL 5000 to get itself set up, it is a good idea to make sure the CAPS-LOCK is on, as ANVIL 5000 occasionally fails to be case-insensitive. Also, remind yourself to type very slowly, as ANVIL 5000 has an extremely moronic input-handler.

When it finally gets ready, ANVIL 5000 will ask you for a "PART FILE". Unless you wish to continue work on an existing part, slowly enter the name NASCAP <CR>. This is an empty Part File which we have provided for the user. You must call the Part File "NASCAP" the first time you wish to input an object. Then you must rename it to avoid writing over the Part File. See Section 4.3.1.1.

A grid will appear on the screen. If the user has called up an existing Part File, then that saved Part will also appear on the screen. If the user entered NASCAP/GEO, the grid will be empty.

If the hardware setup includes a tablet, the user will be prompted to "SELECT FUNCTION" from the "objdef" overlay. Otherwise, the ANVIL 5000 Main Menu will appear on the screen. ANVIL 5000 is now ready to go.

4.2.3 Getting Started: A Plan of Attack

We recommend that the user first skim through the rest of this chapter, and then begin trying to define building blocks right away. It is important to have a little familiarity with the Graphics Interface before the user starts to define the actual objects that will be run through NASCAP/GEO or POLAR 1.1. This can probably be accomplished in one or two sittings.

The user should understand the process of deleting entities in case mistakes are made, which is certain to happen. Planning ahead to use levels when developing the model is necessary to delete entities easily. See Sections 4.3.2.2 and 4.3.6.

As was mentioned before, follow the instructions on the screen, and if something does not look right or familiar, don't panic, just look it up in this chapter. Remember that it is possible in most cases to go back to the previous menu or prompt by hitting a ([) key.

4.3 OBJECT MANIPULATION

This section deals with the main ANVIL 5000 functions that the user will use in defining and manipulating objects. It contains both step by step instructions and important discussions to facilitate the use of ANVIL 5000.

4.3.1 Filing and Merging

This section discusses two commands, "File Part" and "Merge." The former will be used very often and it is important that the user be familiar with how to file Parts in order to save the work done. The latter is a less used command/technique, but it is discussed here because of its nature.

4.3.1.1 File Part Command

One of the first things the user will want to be able to do is to save the current "PART" and be able to exit the ANVIL 5000 Graphics Interface.

The "File Part" option allows one to do these. When selected, it will ask, "FILE?" If a "Y" is entered, the current Part with the latest changes will be saved. The current Part, named <name>, will be saved onto a <name>.PRT file.

If an "N" is entered after it asks "FILE?", it will not save the current Part.

ANVIL 5000 will then ask, "TERMINATE?" If an "N" is entered, ANVIL 5000 will allow the user to continue working on the current Part, whether the Part has been Filed or not.

ANVIL 5000 will put the user back where he was before "File Part" was executed, and the "SELECT OPTION" message should appear on the screen. Problems may occur when using "Y" or "N" from the tablet in this case. The keyboard should be used.

To quit working on the current Part, whether it has been Filed or not, enter a "Y" when it asks "TERMINATE?"

If a "Y" was entered after "TERMINATE?", ANVIL 5000 will then ask, "NEW PART WANTED?" Enter an "N" to quit ANVIL 5000 altogether.

To continue work on an existing Part File, or begin a new one, enter a "Y" when it asks "NEW PART WANTED?" Then, when ANVIL 5000 queries for a "PART FILE" name, the user should type in the name of the Part to be called up on the screen. Problems may occur when using "Y" or "N" from the tablet in this case. The keyboard should be used.

4.3.1.2 Merge Command

"Merge" is used to combine two different Part Files. This is useful if the user is working on objects with many surfaces, and wishes to split up the task into two or more Parts.

Using the "Merge" option successfully requires a little bit of planning. When a Part is Merged into another one, all of the surfaces transferred will have the same coordinates as when they were originally defined. Therefore, if one Part is to "fit" into another, the coordinates have to be the correct ones on both parts as is needed for the case.

NOTE About Block and Surface Numbers

When a Part is Merged into another one, it is important that they not have duplicate "names", SURF(i). Use the "List Entities" and INITBS commands to view and set part names. If the user has been Deleting Entities, or Manipulating them in some way, the same care with Surface #'s should be taken as when a building block is defined. (See 4.3.2, Entities and Surfaces, for details.)

Using Merge

When "Merge" is selected, the user will be prompted to "ENTER NAME OF PART TO BE MERGED". At this point, the Part to be merged should be entered. To cancel the Merge command, enter a ([) instead.

After the file has been specified, ANVIL 5000 displays the menu:

ORIGIN MODE

1. SCREEN POSITION
2. ENTER COORDINATES
3. EXISTING POINT

For most cases, the user should pick "2". ANVIL 5000 will follow with the coordinates for the origin. After all of the values are set to 0, as shown below, the (]) should be hit to continue:

1. XT ORIGIN = 0
2. YT ORIGIN = 0
3. ZT ORIGIN = 0

After the origin has been specified, the following list of data will be displayed,

1. FROM LEVEL= 0
2. TO LEVEL = 0
3. BY INC = 1
4. LEVEL MOVE= 0
5. OR 1 LEVEL= -1

To make sure the whole Part is transferred, the user should indicate that the levels go FROM 0 TO 1000. THE LEVEL MOVE MUST BE = 0 to ensure the rest of the ANVIL 5000 Graphics Interface interprets all objects correctly.

Enter each value, followed by a <CR> to go to the next one. To make a correction, type in the number of the choice to be changed (1 through 5), followed by the new value. Before the (]) key is hit, the user should make sure the values are:

1. FROM LEVEL= 0
2. TO LEVEL = 1000
3. BY INC = 1
4. LEVEL MOVE= 0
5. OR 1 LEVEL= -1

ANVIL 5000 will then display a "THINKING" sign until it is done Merging the Parts.

Other Options

The procedure described above is the simplest method to Merge two files, but excludes several options provided by ANVIL 5000. If the user wishes to use these, he should

refer to the ANVIL 5000 Manual, Section 6.1.4, on different "Merge" options.

For all the applications where a large or complicated object is designed using two or more Parts, and the user wants to Merge the Parts completely at some point, the above procedure will always work.

4.3.2 Listing and Deleting Entities and INITBS

This section discusses three important commands: List Entities, Delete Entities, and INITBS. There are two types of entities: one marking the start of a new building block (a block point) and those which define the surfaces of the building block. Every object created by ANVIL consists of a block point followed by a varying number of surface definitions.

4.3.2.1 List Entities Command

List Entities allows the user to see a list of all or part of the entities that comprise the various building blocks which have been defined. The entities carry the name of the surface which they define, in the form SURF(n), where n is a number.

When List Entities is selected, ANVIL 5000 will ask "SELECT ALL DISPLAYED?" To get a complete list of all of the surfaces presently defined on the screen, enter a "Y".

To list specific entities, "SELECT ALL DISPLAYED" should be followed by an "N". The SCREEN SELECT MODE Menu will then appear. This is shown below:

SCREEN SELECT MODE

1. SINGLE
2. CHAIN
3. INSIDE REGION
4. OUTSIDE REGION
5. RANGE OF DISPLAYED LEVELS
6. ALL DISPLAYED

Option "6" is equivalent to answering yes to the previous question about having all of the displayed entities listed. This choice leads directly to the list of all the surfaces presently defined on the screen.

Option "5" allows the user to have a number of surfaces listed according to their Level Numbers. If the user has incorporated this more advanced technique into his object definition procedure, this would be a plausible and self-explanatory choice. (See Section 4.5 for a complete discussion on the use of Level Numbers.)

Options "3" and "4" entail selecting a number of surfaces to be listed by picking a region on the screen. This might be useful, depending on the particular case, but could also lead to some confusion if there is three-dimension clutter on the screen. These are good options if the object being defined has specific, spread-out sections consisting of different building blocks.

Option "2" is not applicable within the use of ANVIL 5000 for object definition.

Option"1" leads to the SINGLE SELECT FORM Menu. From here the user can specify which entities are to be listed by choosing "4" (ENTER NAME), and then entering any number of

surfaces which have been defined, with the name SURF(n). From the SINGLE SELECT FORM Menu, the user can also choose "1" (SCREEN SELECT). ANVIL 5000 will then prompt for the entities to be selected b pointing to them on the screen. Any number of displayed, defined surfaces may be chosen in this manner.

To indicate that all the entities desired have been selected (from Options 1, 3 or 4), a <CR> or (]) should be entered. This leads back to the SCREEN SELECT MODE Menu. From here, the user can get the actual desired list by entering a (]). Selecting any of the options now permits the user to repeat from the top the process of selecting which entities are to be listed.

What is in the List of Entities?

Once the user gets to the actual list, ANVIL 5000 will display the Name, Type and Sequence Numbers of all the specified entities. The "Name" will be of the form SURF(n), and these will be in the order that they were defined. The "Type" reflects the kind of entity that it is; a "1" signifies that the entity is a block point, and any other number (e.g. 19, 21) signifies that it is a surface. The "Sequence Number" is appointed by ANVIL 5000, and will probably not be of use to the user.

Within each page of the list, the user will be asked to "CONTINUE?". This refers to going on to other pages of the list of entities if such exist. Typing a "Y" will continue the list, or exit the list if there are no more entities. Entering an "N" takes the user out of the list.

After the user is done looking at the list, ANVIL 5000 will put up an EXTENT OF ENTITY SELECTION Menu, a "1" (SELECT FROM ALL) should be entered. Options "2" and "3" will be of no use. To go on to the SELECT FUNCTION prompt, a (]) should be entered.

4.3.2.2 Delete Entities Command

Delete Entities allows the user to delete any number of entities which have been defined with a GRAPL program. The user specifies which entities are to be deleted by name (e.g. SURF(i)), or by pointing them out on the screen.

When Delete Entities is selected, ANVIL 5000 will prompt the user to ENTER ENTITY NAME. For most applications in the ANVIL 5000 - NASCAP/GEO Interface, the entities will be surfaces. Thus, the user will enter SURF(i), where i is the surface number to be deleted.

After SURF(i) has been entered, a <CR> or (]) should be hit. ANVIL 5000 will then display ENTER ENTITY NAME again. To delete more than one surface, the user should enter the next entity name. The user must also delete the block point, with TYPE #=1, associated with the surface being deleted. When all the entities to be deleted have been entered, ENTER ENTITY NAME should be followed by a (]). ANVIL 5000 will delete the entities specified. See Section 4.3.6 for more hints on deleting entities.

File Part Before Deleting Surfaces

To abort the Delete Entities function, enter a ([) when ENTER ENTITY NAME is displayed on the screen. After surfaces have been deleted, however, there is no way to get

them back. Therefore we recommend that if the user is not completely sure of which entities are to be deleted, and even if he is, that the Part be Filed before Delete Entities is executed.

4.3.2.3 INITBS Command

INITBS is used to initialize the next Block number and Surface Number to be used by a GRAPL program.

The Block # is the number of the next building block to be defined. For example, if the user has defined two RECTANs and two PLATEs, and now wants to define a WEDGE, the WEDGE would be Block #5.

The Surface # is the number of the next surface available for the next building block. Surface #n is synonymous with the entity named SURF(n).

Each Block # will have at least two surfaces associated with it. The first surface, SURF(n), is the block point, and the second surface, SURF(n+1), is the first surface defined of the building block. The next surface defined, if there is another, would be SURF(n+2), and so on.

For example, if a RECTAN with six surfaces is defined, and the first Surface # used is n, the RECTAN will be defined by SURF(n) through SURF(n+6).

When to use INITBS

If the Block and Surface #'s are not initialized with INITBS, ANVIL 5000 will automatically assume that the next available Block and Surface numbers after the previously

used ones are to be used. Thus, if one were to define objects without ever making a mistake, or more importantly using "Merge," there would be no need for using the INITBS Command.

The occasion in which it is absolutely necessary to use INITBS is after "Merge" is executed (see Section 4.3.1). With the newly merged file, if a GRAPL program is run, ANVIL 5000 will assume the Block # and Surface # to be used are the next ones from the part that the object was "Merged" into. Hence, if a part is "Merged" into an empty part file, and if a GRAPL program is executed without using INITBS, ANVIL 5000 will want to begin defining surfaces with Block #1, Surface #1.

Thus, after a "Merge", the user will want to INITBS so that surface definition begins at the correct spot. Otherwise, the user risks defining surfaces over already existing surfaces, which could be disastrous.

Other Recommended Times to Use INITBS

There are other times when the user may opt to use the INITBS function. A couple of suggestions are mentioned here.

When "Delete Entities" is used, ANVIL 5000 still saves the Block # and Surface # of the deleted building block. Thus, when the user goes to execute the next GRAPL program, ANVIL 5000 will want to use the next available Block # and Surface # after the previously defined building block, whether that particular building block still exists or has been deleted.

If the user is far along in a many building block object definition, and the need to delete a building block from somewhere in the middle arises, INITBS should probably NOT be used. There is nothing wrong with leaving a "hole" as far as Block #'s and Surface #'s go.

However, if the user is just beginning a part file, and the need to delete the latest building block defined arises, INITBS can be used to clear things up. It might be confusing to have a number of Surface #'s missing when there are not that many surfaces presently defined, and it would be a simple task to figure out which Block # and Surface # to "Initialize" when there are few building blocks defined.

INITBS: A Last Word

The most important thing to remember is to INITBS to the correct values after "Merge" has been executed. Otherwise, if the user is not careful, building blocks could be defined over existing ones.

The user should also remember that no harm will come from "Deleting Entities" and leaving "blank" spots in Block #'s and Surface #'s, other than possible confusion if one does not remember what happened before.

4.3.3 Writing to an IGES File

This section describes the IGES Command. This is how Parts defined with the ANVIL 5000 Graphics Interface have to be saved when they are finally going to be translated to a NASCAP/GEO objdef file.

In order to go to the next step of the ANVIL 5000 - NASCAP/GEO Interface, the Part File defined with ANVIL 5000

must be saved onto an IGES file. IGES files will be designated as <name>.TAP files on the user's directory.

To write an IGES file, choose the IGES Command when the "SELECT FUNCTION" message is at the top of the screen. The user will then be asked,

THIS FUNCTION USES THE CURRENT WORK
SPACE - FILE CURRENT PART?

To indicate that the current Part is to be translated into an IGES file, the user should enter a "Y" or (]) to continue.

To cancel the IGES Command now or anywhere before the file has been written, hit the ([) key once to return to the "SELECT FUNCTION" message; or once to go up one menu and then a ([) again to return to the "SELECT FUNCTION" message.

ANVIL 5000 will then prompt, ENTER OUTPUT FILE. The user should slowly type in the name that the IGES file will have, followed by a <CR> or (]) to continue. Normally, the name is your "part" name with the explicitly supplied extension .TAP.

ANVIL 5000 will then prompt, ENTER NAME OF FILE TO SAVE. The name of the current Part should be entered, without the ".TAP" extension, followed by a <CR> or (]) to continue.

The user will be shown a menu displaying a number of options about initializing the IGES file. In most cases, there will be no need to enter anything, so the user can hit a (]) to go on. Anything entered at this point will be used in the IGES file's header and is ignored by the CAE Tools.

The message "THINKING" will appear on the screen while the ANVIL 5000 Graphics Interface is writing the IGES file.

When it is done, ANVIL 5000 will prompt again, "ENTER OUTPUT FILE". This is in case the user wants to write another IGES file at this point. To do so, enter the name of the file, and follow the instructions.

If the user does not have another "file to save", a (]) should be hit to go on. The question, "NEW PART WANTED?" will follow. To quit ANVIL 5000 altogether, enter an "N".

If the user wants to call up another Part at this point, "NEW PART WANTED?" should be followed by a "Y", and ANVIL 5000 will start at the very beginning (i.e. "ENTER PART FILE NAME", etc.).

4.3.4 Grid Options and More

This section discusses the grid and coordinate system setup, and some of the different options the user has of controlling the display of the grid.

Coordinate System and Grids

The coordinate system for the ANVIL 5000-NASCAP/GEO interface is $1<X<33$, $1<Y<17$, $1<Z<17$. Since NASCAP/GEO requires Z to be the long dimension, the IGES-Objdef translator will perform the rotation X->Z, Y->X, Z->Y. Remember that surface may not lie in the boundary planes of the primary grid. For POLAR 1.1, they may not touch the boundary planes. The analysis codes will not be able to

convert objects into their internal representation if these restrictions are violated.

The grid may be suppressed for less cluttered viewing of the object. To do this from the ANVIL 5000 Main Menu, select "Set Modals," select "Grid," and then select "Grid On, No Display."

ANVIL 5000 uses an "active" grid. This means that when a coordinate is selected on the screen, ANVIL 5000 automatically translates it to the nearest grid intersection.

4.3.5 Tablet Versus No Tablet

The "objdef" tablet overlay has been designed to simplify the definition of NASCAP/GEO objects with ANVIL 5000. However, not all ANVIL 5000 operations can be done from the tablet, and not all hardware setups include tablets. Table 1 lists the tablet entries and the equivalent keystrokes in menu mode.

To switch from tablet mode to menu mode, click the small "Return to menus" box on the top row of the tablet.

To switch from menu mode to tablet mode, enter the keystrokes

```
ctrl-F 1 0 4
```

where "ctrl-F" is to go to the ANVIL 5000 Main Menu. "1" is for the System Modals Menu, "0" is to choose the Keyboard/Tablet Input Mode, and "4" sets the Full Tablet into effect.

YES		REJECT		NO		Operation Complete		ZOOM		Regenerate Display		Top Level Return	Change Oper Mode	Delete Last Entry	Change Depth	Data Entry Key In	[illegible] Entity Select	Return [illegible] Menu [illegible]	Tablet Entity Select
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	=
A	B	C	D	E	F	G	H	I	J	K	L	M	Space		0	1	2	3	4
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Return		5	6	7	8	9

RECTAN	OCTAGO	SLANT	Change View	File Part
WEDGE	QSPHER	ASLANT	Multiple Views	Rename Part
TETRAH	FIL111	ATET	8 Views	Merge
PLATE	PATCHR	APLATE	Continue GRAPL	IGES
BOOM	PATCHW	INITBS	List Entities	Delete Entities

View	Norm. Dir.
Front	+
Back	-
Top	+
Bottom	-
Left	-
Right	+

ANVIL-5000/GRAPL Tablet for GEOCAT and POLCAT

S-CUBED

Table 1

Tablet Choices and Menu Keystrokes

Tablet Choice	Menu Keystrokes
RECTAN	ctrl-F 5 2 2 RECTAN
WEDGE	ctrl-F 5 2 2 WEDGE
TETRAH	ctrl-F 5 2 2 TETRAH
PLATE	ctrl-F 5 2 2 PLATE
BOOM	ctrl-F 5 2 2 BOOM
OCTAGO	ctrl-F 5 2 2 OCTAGO
QSPHER	ctrl-F 5 2 2 QSPHER
FIL111	ctrl-F 5 2 2 FIL111
PATCHR	ctrl-F 5 2 2 PATCHR
PATCHW	ctrl-F 5 2 2 PATCHW
SLANT	ctrl-F 5 2 2 SLANT
ATET	ctrl-F 5 2 2 ATET
APLATE	ctrl-F 5 2 2 APLATE
INITBS	ctrl-F 5 2 2 INITBS
Change View	ctrl-F 8 3
Mult. Views	ctrl-f 8 4
8 Views	ctrl-f 8 4 0 5
Continue GRAPL	ctrl-F 5 2 3
List Entities	ctrl-F Shift-9 9 3 1 6
File Part	ctrl-F 4
Rename Part	ctrl-F 6 1 2
Merge	ctrl-F 6 1 7
IGES	ctrl-F 7 6 2
Delete Entities	ctrl-F 3 3 1 4

4.3.6 Advanced Object control: The Use of Levels

Level manipulation is a more advanced technique which can be used to specify different objects. It uses the fact that ANVIL 5000 requires the user to give each surface defined as Level Number. Once mastered, it can be used to facilitate such useful functions as List Entities, Delete Entities and Merge.

When a GRAPL program is executed, the user is prompted to enter a Level Number for each surface defined. ANVIL 5000 then associates each surface with this entered value, and the user can later refer to particular surfaces by taking advantage of this feature. Thus, the user can differentiate between building blocks defined by use of levels.

Using Levels

The most efficient way to make use of levels is to assign each building block or related set of building blocks a different Level Number. If the object being defined is small enough (in number of building blocks), then the user can give each building block a different Level Number, and then be able to refer to each block as a unit. For larger objects, the user can give each section of the object a different level number, and then have these sections available as referable units.

If levels are to be used, special care should be taken to make sure that different surfaces of one building block do not have different Level Numbers. This could make life quite complicated if, for example, the user wanted to delete a RECTAN by referring to it by levels, but left part of it behind because a surface had a different Level Number.

Quite simply, the way to use levels is as stated: the user gives a Level Number to each surface, and then refers to items by stating a range of levels to do a certain action upon. One way we recommend doing it is to have each building block have the same Level Number as Block #. This may get too confusing for parts consisting of many building blocks; for these, since a fair amount of planning will have

to be done anyway, the user can choose to have certain pieces of the object have the same Level Number, as described above.

When the user actually opts to refer to entities by their Level Numbers, the following menu will appear to indicate which levels the user wants to take the action on (this menu can be reached from the "List Entities," "Delete Entities" and "Merge' functions, among others):

1. FROM LEVEL = 0
2. TO LEVEL = 0
3. BY INC = 1

The user should enter these in the typical way, indicating the level numbers in question.

For more detailed information on the use of levels, the user should consult the ANVIL 5000 GRAPL Reference Manual. The description above should suffice for most uses in relationship to the CAE Tools.

4.4 GRAPL PROGRAMS AND BUILDING BLOCKS

This section discusses several important features of the Graphics Interface which will be used time and time again in defining building blocks. We suggest the user read through the material on views, out of screen coordinates, and defining surfaces as background knowledge to the Graphics Interface.

4.4.1 Views of the Object

Changing Views

Views may be changed via the "Change View" tablet choice. Views available by (case-sensitive) name include "FRONT", "BACK", "TOP", "BOTTOM", "LEFT", "RIGHT", "TILT1", "TILT2", and "TILT3".

After the View Name is chosen, the user will be asked about scaling the view. For the first six views, "LAST SCALE USED" should be correct. We have provided two named views, "4207" and "4014", for each of these views. These correspond to the Tektronix 4207 and 4014 respectively, and can be entered as a saved value for the scaling.

For the "TILTi" views, "AUTOMATICALLY MAXIMIZED" is usually the best option.

The "8 Views" tablet choice will draw the six orthogonal views of the object, together with two 45-degree views. ANVIL 5000 figures out its own scaling needed for the "8 Views" selection.

A Few Words of Advice

The key to using ANVIL 5000 effectively is knowing in which view to define the desired building block. The user must learn to visualize from which side an object is being looked at in each view. A cardboard box, with six views of the model posted on the sides, can help visualize the model. It is important to know this so that the correct view will be chosen for defining an object.

This is especially useful as soon as several objects are defined on the screen at once. When things start looking really clustered, the user will want to be able to anticipate the appearance of the next building block and the view in which it must be defined to fit in correctly with the rest of the object.

Change Views Often at First

If not familiar with the GRAPL programs, the user may want to Change Views often at first to gain a better idea of what different building blocks look like in the various planes. Another idea would be to look at an angled view (TILT1, for example) or even all 8 Views at once in order to see the object from different angles.

Changing Views is also the best way to check that the correct coordinates out of the screen, Z1 and Z2, have been specified. (Figure 4.4.1 shows the six orthonormal views and which planes they define)

NOTE About Negative Coordinates on Axes on Certain Views

In three of the views, TOP, RIGHT and BACK, an axis is displayed with negative values (see Figure 4.4.1). They are negative because that particular axis is pointing either to the left or the bottom, while the convention is to have positive to the right or the top.

Thus, in these cases, the minus signs reflect the direction of the axis. These views can still be used, but the user must be aware of this so as to avoid possible confusion.

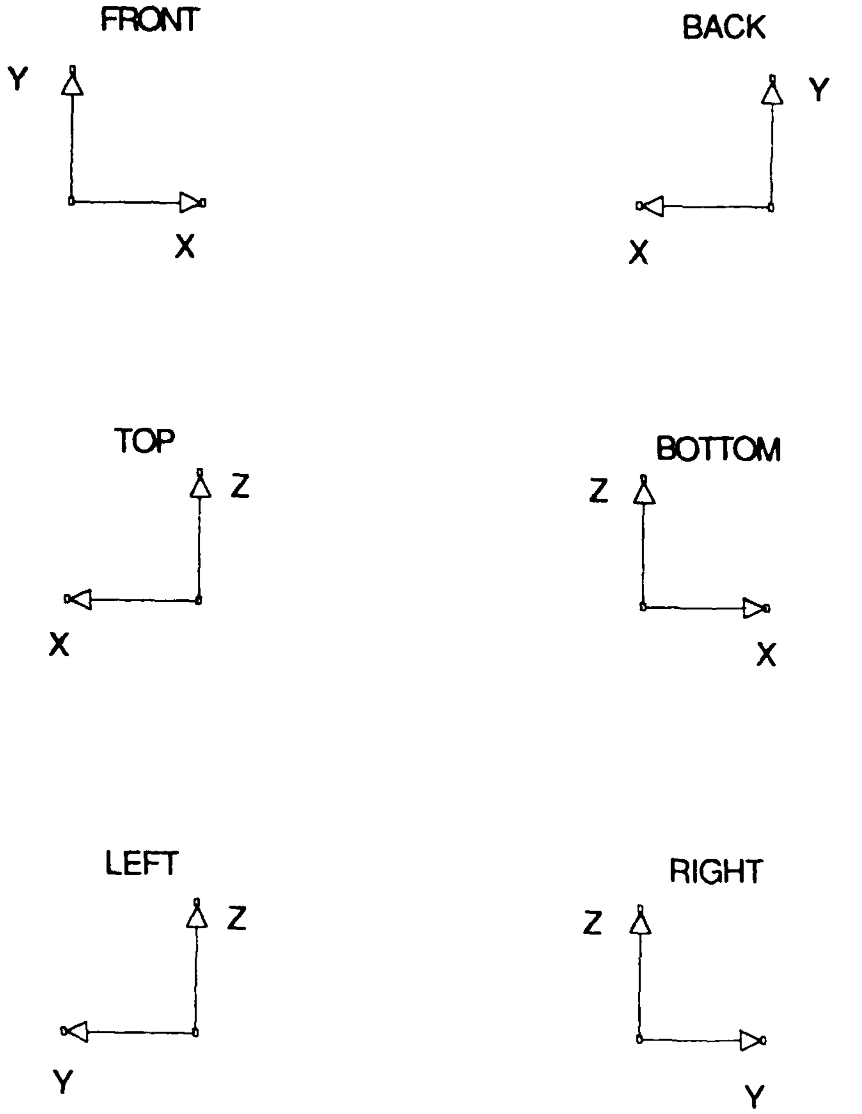

Figure 4.4.1 Orthonormal Views and the Planes they Define

A Word of Warning:

Although Changing Views is an essential part of defining a tool by knowing its limitations as well as its capacities.

Unfortunately, ANVIL 5000 does not differentiate between surfaces that are nearer to you from those that are further away. In other words, if there are two plates on top of each other, one would expect to see the surface facing out of the screen from the top plate, not one of the underneath surfaces. However, this will not necessarily be the case in ANVIL 5000.

Furthermore, even if there is just one plate, with a top and bottom surface defined, one would expect to see the top surface when one is looking at the plate from the top, and the bottom surface when one looks at it from the bottom. Again, not so in ANVIL 5000.

What ANVIL 5000 does is display all the surfaces which are supposed to be on the screen, but with no depth or top-bottom differentiation. It correctly keeps track of the different surfaces, but sometimes does not display them clearly on the screen.

The user should therefore take great care to choose top versus bottom and slanted faces correctly when the building block is first defined. Otherwise it could be a confusing task later to double check different material types on different surfaces using the ANVIL 5000 display.

4.4.2 Out of Screen Coordinates

The "Out of Screen" coordinates are those which cannot be seen in the view from which the building block is defined (i.e. the depth). They are thus entered from the keyboard when ANVIL 5000 prompts the user to do so.

Entering Values

For building blocks which have two "out of screen" coordinates (Z1 and Z2), enter the choice for the first value followed by a <CR> to go to the second one. If a mistake is made and the values have to be changed, enter the number of the coordinate to be changed (1 or 2), followed by the new value.

When both Z1 and Z2 have been correctly entered, hit the end of data key (]). For building blocks which require only one "out of screen" coordinate, a <CR> functions like the (]).

Interrupting a GRAPL Program

In general, the GRAPL programs may be interrupted when they ask for an "out of screen" coordinate. To do this, hit the REJECT ([) key, rather than entering data. The "PAUSE" Menu will be displayed. Choose "1" (File Program).

The user may then, for example, change the view in order to see what coordinates should be entered. The user must then return to the original view before continuing the GRAPL program. To recommence the GRAPL program, make the "Continue GRAPL" choice, and type in the GRAPL program name,

using the keyboard, not the tablet. ANVIL 5000 will continue at the point where the GRAPL program was interrupted. Do not choose the program to be continued from the tablet, as this will start the GRAPL program from the top.

Comments

The order in which Z1 and Z2 are entered does not matter. In other words, if the user enters Z1 = 10 and Z2 = 2, ANVIL 5000 will still define the building block from Z = 2 to Z = 10. For specifics, look up the GRAPL program in question.

Finally, if the user leaves an "out of screen" entry blank, ANVIL 5000 will interpret it as a 0. If the two "out of screen" coordinates have values Z = 0, the object will have no width (i.e. depth), but the same number of surfaces can still be defined as if it had nonzero width.

WARNING: Negative "Out of Screen" Coordinates on LEFT, BACK and BOTTOM Views

To define a building block from one of these three views, the "out of screen" coordinates (Z1 and Z2) must be negative, and entered as such.

The reason for this is simple. As the axes are defined in these views, the third axis goes into the screen (see Figure # 1). Hence, the "out of screen" coordinates must be negative to reflect that the desired values are "into the screen."

4.4.3 Defining Surfaces

For all of the GRAPL programs described in the following sections, 4.4.4 to 4.4.12, the user will be defining one or more surfaces for each building block. All of the surfaces of a particular object need not be defined, depending on the particular need.

In general, it is best to always define the low corner of an object or surface, and then the high corner. The low corner of an object is the corner whose grid coordinates are the smallest.

Getting Started

When the desired GRAPL program is executed, after the coordinates have been specified, ANVIL 5000 will ask,

"BLOCK #n, SURF #m, PROCEED?"

(see Section 4.3.2 (b) if you've forgotten what this means)

If an "N" for no is entered, ANVIL 5000 will abort the GRAPL program and go back to the Main GRAPL Menu. If a "Y" for yes is entered, the user will be ready to define the surface(s) of the building block, with Block #n and Surface #m.

Defining Surfaces

The user will be prompted as to whether he wants to define a particular surface. If an "N" is entered, it will leave that surface undefined and go on to the next one.

If a "Y" is entered, it may ask for specifics as to the direction of the top or bottom, or if the normal is facing the user (see individual GRAPL program descriptions), depending on the building block in question.

Then, the user will be prompted to choose the LEVEL and MATERIAL numbers of the surface. The user should enter each choice, followed by a <CR> to go to the next one. If a mistake is made and the values have to be changed, enter the number of the choice to be changed (1, 2 or 3), followed by the new value.

When the LEVEL and MATERIAL have been correctly entered, hit the end of data key (]). ANVIL 5000 will then draw the surface just defined, and it will go on to the next surface if there is one.

A Word About LEVEL and MATERIAL Numbers

The range of choices for the possible values of LEVEL and MATERIAL number is 1 through 15.

We recommend that the user keep a legend as to which color and number represents which material. This way, if one wants to check the material one has chosen for a particular surface, all one has to do is look at the color displayed on the screen.

Table 4.4.10 (list of colors) lists the colors corresponding to each number. Note that color #1 is black, which the user may find an unsatisfactory choice.

Conductor Numbers

A conductor number between 1 and 15 is associated wth each block. Due to limitations of ANVIL 5000 v 1.1, conductors 1-8 are defined correctly, but conductors 9-15 reappear as 2-8. The system editor can be used to correct this in the Object Definition File.

Table 4.4.10

COLOR VS MATERIAL NUMBER TABLE

COLOR	MATERIAL #
black	1
red	2
green	3
blue	4
yellow	5
magenta	6
cyan	7
white	8
brown	9
olive	10
aqua	11
purple	12
orange	13
pink	14
gray	15

The choice of which LEVEL number entered is not important yet. For now, any integer between one and ten should do.

LEVELs are used for more complicated techniques, allowing for specific differentiation of particular surfaces by distinguishing them according to LEVEL numbers. At this stage, this feature will probably not be needed. See Section 4.3.6.

4.4.4 RECTAN and PATCHR

4.4.4 (a) RECTAN

RECTAN is the GRAPL program which allows the user to define a cuboid or rectangular parallelepiped. The user is thus allowed to define from one to six surfaces.

The cuboid is the simplest building block to define in that it may be defined correctly from any view. The user defines a rectangle in the view in which RECTAN is executed, and then specifies the depth of the cuboid (i.e., the coordinates "out of the screen").

Defining a RECTAN

When RECTAN is selected the user will first define a rectangle by picking two opposite corners on the screen. When this is done, ANVIL 5000 will ask for the out of screen coordinates (Z1 and Z2). Enter these as desired for the depth of the cuboid. Also enter the conductor number.

The user will then get a chance to define the various surfaces, starting with the left side. After this surface has been defined, ANVIL 5000 will ask "DITTO ALL SURFACES?". To indicate that all of the surfaces of the RECTAN are to have the same LEVEL and MATERIAL numbers as the left surface, enter a "Y". If an "N" is entered, ANVIL 5000 will go to the next surface. For each surface the user opts to define, ANVIL 5000 will prompt for the COLOR, LEVEL and MATERIAL numbers to be entered.

When this process is completed, the user will be done defining the cuboid, and be returned to the Main GRAPL Menu.

Comments/Hints

As is mentioned above, this GRAPL program is very straightforward. The cuboid represents the simplest type of building block, and the user should be familiar with RECTAN before other types of shapes are attempted.

4.4.4 (b) PATCHR

A PATCHR is very similar to a RECTAN. Its shape, and the way in which it is defined, are identical. The only difference is that in PATCHR the user does not have the option to do a "DITTO ALL SURFACES". Because of the nature of the PATCHR, we left this option out since, in most cases, all six surfaces will not be defined.

PATCHR is used to "patch up" surfaces, usually with the purpose of having a different material and/or conductor on a portion of another object. The user will usually only have to define one or two surfaces of the PATCHR, depending on the need.

NASCAP/GEO and POLAR 1.1 require that a PATCHR be defined inside of another building block. Thus, PATCHR should not be used instead of RECTAN, but only when the user needs to "patch up" an existing object. PATCHR should not be used on plates. Rather, several plates should be used.

4.4.5 WEDGE and PATCHW

4.4.5 (a) WEDGE

WEDGE is the GRAPL program which allows the user to define a wedge. A wedge is a cube sliced diagonally in

half. The user is thus allowed to define from one to five surfaces; four sides and the face (slanted surface). (See Figure 4.1.2 (b).)

A wedge is defined on the view in which it forms an isosceles, right angle triangle. Thus, the user will first define the triangle, and then specify the depth of the wedge (i.e. the coordinates out of the screen).

Defining a WEDGE

When WEDGE is selected, ANVIL 5000 will ask if the WEDGE appears as a triangle. If an "N" is entered, it will abort the WEDGE program, and go back to the Main GRAPL Menu. If a "Y" is entered, it will continue, and ask the user to indicate the coordinates of the triangle on the screen.

After the triangle has been defined, it will ask for the Z1 and Z2 coordinates. Enter these as desired for the wedge. Also, enter the conductor number.

The user will then get a chance to define the various surfaces, starting with the slant. For each surface the user opts to define, ANVIL 5000 will prompt for the LEVEL and MATERIAL numbers to be entered. The slant of the WEDGE must be defined to translate properly.

When this process is completed, the user will be done defining the wedge, and be returned to the Main GRAPL Menu.

Comments/Hints:

The user should make sure the plane (i.e. view) in which the triangle of the WEDGE is seen and defined is the

appropriate one for the particular object. Then, from all the four other orthonormal views, the user will see a rectangle: the superpositioning of the slanted face and one of the sides (provided these were defined, of course).

Thin Triangles

Finally, there is a special kind of shape which may be defined with the WEDGE program: the thin triangle. A thin triangle is the name given to a WEDGE in which the only surface defined is one of the two triangular sides. The user should remember that thin triangles are only accepted by NASCAP/GEO, NOT by POLAR 1.1

4.4.5 (b) PATCHW

A PATCHW is very similar to a WEDGE. Its shape, and the way in which it is defined, are identical.

PATCHW is used to "patch up" surfaces, usually with the purpose of having a different material on a portion of another object. The user will usually only have to define the slanted surface and one or two other surfaces of the PATCHW, depending on the need.

NASCAP/GEO and POLAR 1.1 require that a PATCHW be defined inside of another building block. Thus, PATCHW should not be used instead of WEDGE, but only when the user needs to "patch up" an existing object.

4.4.6 TETRAH

TETRAH is the GRAPL program which allows the user to define a tetrahedron. The user is thus allowed to define

from one to four surfaces. These are the three sides which are isosceles, right-angled triangles, and the unique face opposite the right angle corner.

A tetrahedron in any orthonormal view forms an isosceles, right-angled triangle. Thus, the user will first pick the right angle corner of the tetrahedron (a point on the screen and Z1), then defines the triangle, and the direction of the normal. It is important to be able to visualize what these entries are selecting. (See Figures 4.1.2 and 4.1.3.)

Defining a TETRAH

When TETRAH is selected, ANVIL 5000 will ask the user to "SELECT [the] RIGHT ANGLE CORNER". This point is picked on the screen. The third coordinate, the one out of the screen, is then entered as Z1 when ANVIL 5000 prompts, "Z-DIST IN FRONT OF SCREEN". This defines the right angle corner of the tetrahedron.

The user is then prompted to choose the "HORIZONTAL CORNER". This point can be either to the right or left of the right angle corner. Once the horizontal point has been picked, ANVIL 5000 will ask, "IS VERTICAL DIRECTION UP?" If a "Y" is entered, ANVIL 5000 will automatically place the vertical point at the appropriate spot above the right angle corner (remember that the sides of the triangles have the same lengths in a tetrahedron). If an "N" is entered, the vertical point appears below the right angle corner.

Once the triangle has been defined, ANVIL 5000 will ask, "IS NORMAL DIRECTION TOWARDS YOU?" In most cases, the user will want to define the tetrahedron in a view in which

the normal is towards him. Thus, a "Y" should be entered (see Comments/ Hints below for a discussion on directions of the normal).

The user will then get a chance to define the various surfaces, starting with the face of the tetrahedron. After the face has been defined, ANVIL 5000 will ask "DITTO ALL SURFACES?" To indicate that all of the surfaces of the tetrahedron are to have the same LEVEL and MATERIAL numbers as the face, enter a "Y". If an "N" is entered, ANVIL 5000 will go to the next surface. For each surface the user opts to define, ANVIL 5000 will prompt for the LEVEL and MATERIAL numbers to be entered.

When this process is completed, the user will be done defining the tetrahedron, and be returned to the Main GRAPL Menu.

Comments/Hints

The TETRAH can be a complicated building block because the user will see a triangle from any view in which the tetrahedron is defined. From any orthonormal view, the user always sees a side and the face superimposed (provided these were defined, of course). It is important that the user try to keep track of which surface is the face as opposed to the other three sides. Perhaps Changing Views often, or looking at a TILTed view, would be a good idea when the user is first getting acquainted with TETRAH.

As was mentioned above, the user will want to define the normal of the face to point "out of the screen" in most cases. To see the difference, try it both ways, and check several views.

At first, a good way to differentiate between the sides and the face when the TETRAH appears on the screen would be to define a tetrahedron which has one or two sides NOT defined. Then look at it from different views, and it should become clearer.

4.4.7 PLATE

PLATE is the GRAPL program which allows the user to define a thin plate. A PLATE is essentially a RECTAN with zero width. The user is thus allowed to define one or two surfaces (see diagram).

The thin plate is a simple building block to define because it will look like rectangle from two orthonormal views, and like a line in the other four. The user defines a rectangle in the view in which PLATE is executed, and that's it.

Defining a PLATE

When PLATE is selected, ANVIL 5000 will ask if the PLATE appears as a rectangle. If an "N" is entered, it will abort the PLATE, and go back to the Main GRAPL Menu. To continue with the PLATE, enter a "Y".

The user will then define a rectangle by picking two opposite corners on the screen. The lowest corner should be chosen first. When this is done, ANVIL 5000 will ask for the "out of screen coordinate". This is the distance "out of the screen" at which the thin plate will be defined.

The user will then get a chance to define the two surfaces, starting with the bottom one. ANVIL 5000 will ask if this is in the "NEGATIVE DIRECTION IN MODEL SPACE?" The user should answer "Y" to make the normal of the bottom surface point "into the screen". Thus, the "top" surface, the "POSITIVE DIRECTION IN MODEL SPACE", will point "out of the screen" (see Comments/ Hints below for a discussion on top vs. bottom).

For each surface the user opts to define, ANVIL 5000 will prompt for the LEVEL and MATERIAL numbers to be entered. When this process is completed, the user will be done defining the plate, and be returned to the Main GRAPL Menu.

Comments/Hints

As is mentioned above, this GRAPL program is very straightforward. A PLATE should be thought of as nothing more than a RECTAN with no width.

We recommend that the user make the bottom surface have a normal "into the screen", and the top surface a normal "out of the screen." This is of course only a convention, but it should avoid confusion since it is the standard way to define top and bottom. It may be done the other way if the user finds the need to do so. Note that these normals are reversed in BACK, BOTTOM, LEFT views.

Either way the user opts to define the positive and negative directions in the model space, consistency should be the rule. In most cases, there is no need to defy the convention described above.

Because of the superpositioning of the top and bottom surfaces by ANVIL 5000, it is sometimes hard to tell the two surfaces apart after they have been defined. Thus, we suggest that the user take care to specify them correctly when the building block is first defined, to avoid possible confusion later on. This is of course a moot point if the top and bottom surfaces have the same material, but a very important point if only one surface is defined, to assure that it is the desired one.

4.4.8 BOOM

BOOM is the GRAPL program which allows the user to define a thin cylinder. The user defines a vertical or horizontal line, specifies the radius of the BOOM, and ANVIL 5000 draws the cylinder around the line. The user thus specifies only one surface.

Defining a BOOM

When BOOM is selected, ANVIL 5000 will ask if the BOOM appears as a line. If an "N" is entered, it will abort the BOOM, and go back to the Main GRAPL Menu. To continue with the BOOM, enter a "Y".

The user will then define a line by picking two points on the screen. When this is done, ANVIL 5000 will ask for the "out of screen" coordinate. This is the distance "out of the screen" at which the BOOM will be defined.

ANVIL 5000 will then prompt the user to enter the radius of the BOOM. This value is taken in grid units.

When this is done, the user will have a chance to define the surface LEVEL and MATERIAL numbers.

When this process is completed, the user will be done defining the BOOM, and be returned to the Main GRAPL Menu.

Comments/Hints

BOOMS are simple building blocks to define. The user should remember that BOOMS ARE ONLY ALLOWED ON NASCAP/GEO, NOT ON POLAR 1.1. A boom must be defined along a grid line.

Also, as explained in section 4.1.3, the BOOM is the only building block which may be defined to go beyond the inner 33 by 17 by 17 grid. There is no special consideration in the definition process to extend a BOOM outside the grid. In some cases, the scaling of the view may have to be adjusted (see section 4.4.1).

4.4.9 SLANT

SLANT is the GRAPL program which allows the user to define a slanted thin plate that lies along one axis, and at 45 degrees to the other two. A SLANT should be thought of as the face of a wedge, but with a top and bottom surface. The user is thus allowed to define one or two surfaces.

The user defines the slant from one of the views in which it forms a 45 degree line with the axes in that plane. From the other four orthonormal views the slant will look like a rectangle.

Defining a SLANT

When SLANT is selected, ANVIL 5000 will ask if the SLANT appears as a 45 degree line. If an "N" is entered, it will abort the SLANT, and go back to the Main GRAPL Menu. To continue with the SLANT, enter a "Y".

The user will then define the 45 degree line by picking two points on the screen. If the two points do not define a 45 degree line, ANVIL 5000 will continue to prompt the user to choose the "OTHER END" until this is done correctly. After the line is drawn, ANVIL 5000 will ask for the "out of screen" coordinates (Z1 and Z2). Enter these as desired for the width of the slanted thin plate.

The user will then get a chance to define the two surfaces. ANVIL 5000 prompts the user to specify the top vs bottom surface by showing with an arrow which is which. The user can be explicit about the top and bottom surfaces if it is needed. If this is not desired, the user can simply say "Y" when ANVIL 5000 asks if the surface specified on the screen is the top or bottom one (see Comments/Hints below). For each surface the user opts to define, ANVIL 5000 will prompt for the COLOR, LEVEL and MATERIAL numbers to be entered.

When this process is completed, the user will be done defining the plate, and be returned to the Main GRAPL Menu.

Comments/Hints

It is impossible to distinguish a SLANT from a PLATE in the four orthonormal views in which it is shown as a rectangle, so be careful. On the other two views where it

forms a 45 degree line and on the TILTed views, however, the difference is quite clear. Once again, changing views is recommended and often times necessary.

The choice of TOP and BOTTOM follow the rules discussed in Section 4.1.8. Also, each surface should have the correct material. This can be accomplished by observing the arrow displayed on the screen to differentiate between the top and bottom surfaces.

Because of the superpositioning of the top and bottom surfaces by ANVIL 5000, it is sometimes hard to tell the two surfaces apart after they have been defined. Thus, we suggest that the user take care to specify them correctly when the building block is first defined, to avoid possible confusion later on. This is of course a moot point if the top and bottom surfaces have the same material, but a very important point if only one surface is defined, to assure that it is the desired one.

4.4.10 OCTAGO and QSPHERE

4.4.10 (a) OCTAGO

OCTAGO is the GRAPL program which allows the user to define a right octagonal cylinder. The user is thus allowed to define from one to three surfaces; the octagonal top and bottom surfaces, and the circumferential surface.

An OCTAGO is defined on the plane in which it forms the actual octagon (i.e. the eight sided polygon). The user first defines this octagon, chooses the desired side length, and then specifies the depth of the right octagonal cylinder (i.e. the "out of screen" coordinates).

Defining an OCTAGO

When OCTAGO is selected, ANVIL will ask if the OCTAGO appears as an octagon. If an "N" is entered, it will abort the OCTAGO, and go back to the Main GRAPL Menu. To continue with the OCTAGO, enter a "Y".

The user will then define an octagon by picking a point on the left side, a point on the right side, and one on the bottom side. ANVIL will then display an octagon and ask, "IS THIS OK?", refering to the shape of the octagon it has chosen. If the lengths of the sides look OK, answer "Y". To make the sides longer or shorter, enter an "N", and then hit "Y" or "N" when ANVIL prompts, "MAKE SIDES LONGER?" or "MAKE SIDES SHORTER?", as needed. To go on when the octagon has the desired shape, answer a "Y" when ANVIL asks "IS THIS OK?"

When this is done, ANVIL will ask for the "out of screen" coordinates (Z1 and Z2). Enter these as desired for the depth of the octagon.

The user will then get a chance to define the various surfaces, starting with the bottom and top, and then the circumferential surface. For each surface the user opts to define, ANVIL will prompt for the LEVEL and MATERIAL numbers to be entered.

After the circumferential surface has been defined, the user will be directed to "FILE; SWITCH VIEW; CONTINUE? This tells you that the GRAPL program will execute a PAUSE, in order that you can resume it in another view. This is necessary in order for a GRAPL program to define the circumferential surface. When this point has been reached, respond "y" to the query. You will now see a message "I NEED Z1 RIGHT?". Since ANVIL cannot draw the circumferential surface while it is perpendicular to the screen, a new view is necessary. The new view should have the low-valued circle at the right of the screen. In the case of octagons defined in the FRONT view, the RIGHT view is appropriate. For the other views, see Table 4.4.10. Respond "y" to obtain the GRAPL PAUSE menu. Choose "1" (FILE) and switch to the desired view. then choose "Continue GRAPL" on the tablet and type "OCTAGO" on the keyboard. Respond "n" (Z1 is not at left) and then "y" (Z1 is indeed at right). The circumferential surface should now appear on the screen.

When this process is completed, the user will be done defining the octagon, and be returned to the Main GRAPL Menu.

Table 4.4.10

Defined On	Change To
FRONT	RIGHT or TOP
BACK	RIGHT or TOP
RIGHT	BACK
LEFT	BACK

Comments/Hints

The description above may make the OCTAGO sound more complicated than it really is. This is one of those building blocks which will seem quite straight forward after the user defines one or two octagons, so we recommend that this be done.

4.4.10 (b) QSPHERE

QSPHERE is the GRAPL program which allows the user to define a quasi-sphere. A quasi-sphere is very much like an OCTAGO in that an octagon with a top and bottom is defined, but instead of the cylindrical surface, a QSPHERE approximates the shape of a sphere. Thus, the user defines only one surface (see diagram).

Like an OCTAGO, the user first defines an octagon (i.e. the eight sided polygon). Then the user specifies the "back" of the QSPHERE, in terms of the "out of screen" coordinate, and ANVIL knows where the "front" of the quasi-sphere is since in defining the octagon, the diameter was implied.

Defining a QSPHERE

When QSPHERE is selected, the user will first define an octagon by picking a point on the left side, a point on the right side, and one on the bottom side. ANVIL will then ask, "IS THIS OK?", refering to the shape of the octagon it has chosen. If the lengths of the sides look OK, answer "Y". To make the sides longer or shorter, enter an "N", and then hit "Y" or "N" when ANVIL prompts, "MAKE SIDES LONGER?"

or "MAKE SIDES SHORTER?", as needed. To go on when the octagon has the desired shape, answer a "Y" when ANVIL asks "IS THIS OK?"

When this is done, ANIVL will ask for the "out of screen" coordinate (Z1). This refers to the "back" of the QSPHERE. ANVIL will take Z1 and define the quasi-sphere "forward" from there, where the diameter is the length between the left and right sides of the octagon.

The user will then get a chance to define the one surface. ANVIL will prompt for the LEVEL and MATERIAL numbers of the QSPHERE to be entered.

When this process is completed, the user will be done defining the quasi-sphere, and be returned to the Main GRAPL Menu.

Comments/Hints

The QSPHERE is similar to the OCTAGO, but much simpler in that the user does not need to worry about defining the circumferential surface. Like the OCTAGO, a tricky part may be defining the original octagon, but this can be quickly mastered with a little practice.

4.4.11 FIL111

FIL111 is the GRAPL program which allows the user to define the special shape used to fill in "steps" whose corner line runs at 45 degrees to the grid lines in any axis plane. ANVIL constructs the FIL111 out of a combination of tetrahedra and truncated cubes. The FIL111 has only one surface which the user defines (see Figure 4.4.1a).

A FIL111 is defined in the view in which it forms a 45 degree line (i.e. the plane in which it functions like a "step"). Thus, the user first defines this line, picks the distance "in front of the screen", and then specifies which way the normal of the face of the "step" points.

Defining a FIL111

When FIL111 is selected, ANVIL will ask, "IS CENTERLINE A 45 DEG LINE?". If an "N" is entered, it will abort the FIL111, and go back to the Main GRAPL Menu. To continue with the FIL111, enter a "Y".

The user will then define the 45 degree line by picking two points on the screen. If the two points do not define a 45 degree line, ANVIL will continue to prompt the user to choose the "OTHER END" until this is done correctly.

After the line is drawn, ANVIL will ask for the distance "in front of the screen". This is the point "out of the screen" at which the FIL111 will be defined. Enter this value as desired for the location of the FIL111.

ANVIL will then pick a direction, draw an arrow to show the user, and ask, "IS THIS OUTWARD NORMAL?" There are only two options, and the user should pick the appropriate one for the FIL111 in question.

After one of the two possibilities for the direction of the normal has been specified, ANVIL will ask, "IS OUTWARD NORMAL TOWARDS YOU?" The user will usually define a FIL111 from a view in which the answer to this question is "Y" (See Comments/ Hints below).

When these two queries specifing the direction of the FIL111 have been answered, the surface may be defined, and ANVIL will prompt the user for the COLOR, LEVEL and MATERIAL numbers to be entered.

When this process is completed, the user will be done defining the FIL111, and be returned to the Main GRAPL Menu.

Comments/Hints

FIL111 is one of the more difficult building blocks to visualize. We suggest the user refer to the diagram to get an understanding of how and when this building block should be used.

In most cases, the user will want to define the FIL111 from the view in which the normal of the face points "out of the screen". This is shown in the diagram, and will make more sense once the user has to use a FIL111 as a "step."

This is a toughy, so don't panic and just take your time. If worst comes to worst, the user may have to delete an incorrect FIL111 and try again. There is no crime in this, it just uses up some time. But this is really the only way to become accustomed to this particular building block.

4.4.12 ASLANT, ATET and APLATE

ASLANT, ATET and APLATE are the GRAPL programs which allow the user to define the three types of transparent antenna surfaces (see diagram and section 4.1.4b). Antenna building blocks will be automatically treated as two-sided by NASCAP/GEO, but only one surface is specified by the user

(ANVIL 5000 takes care of the other one since it is by definition the same).

Because of the similarity in the definition process to other GRAPL programs, we simply state the building blocks related to the antenna without repeating the whole process. The only difference is of course that only one surface is defined for each antenna building block

Refer to:

ASLANT is defined exactly in the same way as SLANT.
ATET is defined exactly in the same way as TETRAH.
APLATE is defined exactly in the same way as PLATE.

Comments/Hints

It is impossible to know that a transparent antenna is transparent after it is defined on the screen. ANVIL 5000 does not display any difference between these building blocks and all others. Therefore, an ATET looks just like the face of a TETRAH on the screen, and so on.

This may lead to some confusion at this stage of the model definition process, but the Tools will of course interpret the transparent antenna correctly.

5. ANALYSIS TOOL CONTROL PROGRAMS, GEOCAT AND POLCAT

The control programs are used to translate the IGES 2.0 files created by ANVIL 5000 into the object definition file, edit surface properties and the mesh grid size of the object definition, create rundecks for the appropriate analysis code and its post-processor programs, control the running of the analysis code batch jobs, produce graphics output files for MOVIE.BYU (DYNA-MOVIE), preview calculation results, and maintain data files, rundecks, IGES 2.0 files and graphics output files. The GEOCAT program interfaces with the NASCAP/GEO analysis code, while POLCAT handles the POLAR 1.1 analysis code.

The two control programs are similar enough that it is easiest to describe them both at the same time, making notes of differences when they exist. Functions dealing with the utilities will be the same. Differences arise when the contents of the rundecks or module names are considered.

In order to make this chapter of the User's Manual more useful, the documentation has been organized in the same manner as the program. The modules which appear on the main line menu are also the main section headings. The subheadings of each section are the submodules, and so on. Several introductory sections are provided first which discuss topics of general interest, such as the characters used to maneuver through the menus. The chapter is concluded by a summary of the modified sections of the NASCAP Programmer's Reference Manual and the POLAR User's Manual.

5.1 OVERVIEW

The Control Programs, GEOCAT and POLCAT, contain the following main modules: the object manipulator (line menu item "Object"), a one dimensional surface material/environment interaction analysis spreadsheet ("SuChgr"), the job controller ("NASCAP" in GEOCAT and "POLAR" in POLCAT) the post processing tool ("PostProc"), the file utility ("Files"), and the CAE Tool user environment form ("Misc"). Each module can call the other modules from the main line menu or the exit flag ("Quit") which signals the end of a session. These modules are described in the following sections.

The function of the object manipulator module is to translate the ANVIL 5000 IGES output file into an Object Definition File. The analysis programs read the Object Definition File to define the satellite. This is done in two steps. First the IGES file is translated to an intermediate file which has all of the necessary information for object definition except surface material definitions. The second step entails defining the satellite's surface materials.

The material/environment analyzer contains submodules which search through a list of material and environment libraries, allow the editing of material and environment definitions, save definitions in a user's local database, and perform spreadsheet analysis to give a first estimate of the behavior of a material type in a set of environments.

The job controller is used to create or modify the rundecks for the analysis codes. It provides a simple

editor to define input commands and the facility to read an external file into a rundeck. The controller starts, stops, and monitors the status of analysis runs which are performed in a batch mode.

The post processor provides several tools to view the information in the analysis code data files. Surface data (GEOCAT only) can be viewed in a text mode. A means of paging through the output file created by an analysis run is also available. Plots are generated by keyword input using the job controller rundeck editor and calling the appropriate plot driver from the job control module.

File utilities are available for removing, backing up, and listing files and creating new working directories. System commands can also be issued from the files module.

The CAE Tool user environment form is used to define user-particular constants which control the behavior of the Tools. Generally, the defaults should be sufficient, but the following items may be modified if desired: the default home directory, the directory path, a user sophistication flag, a debugging flag, and the terminal type. The directory path defines the search path, both the directories to be searched and the order in which to do it.

Finally, each module contains a "Quit" flag which is used to terminate a session. This flag should be used to end each use of the Tools smoothly. It resets the terminal to normal operation mode, cleans up any open forms, and deletes any temporary files.

5.2 USING THE USER/SCREEN INTERFACE SCREEN LAYOUT

The user interfaces for GEOCAT and POLCAT both operate in similar fashion and all of this chapter applies equally well to both programs. This chapter explains how the interface works in general terms and should be read by novice users before starting with either GEOCAT or POLCAT. The user interface serves to facilitate moving among the modules within GEOCAT or POLCAT, providing a consistent means of interacting with each module and eliminating much of the tedium of a continuous stream of menus.

The user interface will take over complete control of the display terminal in much the same way as a program editor. The display screen is divided into two parts; a three line command section at the top of the display and a 21 line "forms" window. In the discussion to follow it will be presumed that the display terminal is the standard 24 line by 80 column CRT (Cathode Ray Tube) terminal such as the Digital Equipment Corporation VT100 family of terminals. The user interface requires a terminal with direct cursor addressing and is therefore far less effective on older hardcopy terminals. The interface uses screen handling libraries provided on the host machine and is generally device independent so long as the terminal being used provides some minimum direct cursor addressability.

Each line in the three line command section of the display has a specific purpose. The top line is the line menu section where module or submodule names are provided. Moving the cursor over to any one of the module names and hitting the return key selects that module for execution. The second line is a command line which is generally used to display one line error messages, yes/no prompts, and simple

text input with prompting. The third line is an annunciator line which is always highlighted and often displays annunciators indicating modes or notices to the user that something is in progress.

The lower 21 lines of the display serve as a window onto a potentially large display buffer. The contents of the buffer consist of whole or partial lines of information called fields. The collection of fields in a buffer constitutes a form. The user interface can have a number of forms active at any time which allows the user to quickly switch from module to module. There are four types of fields, two to which the cursor can be moved and two to which the cursor is not allowed to move. The fields to which the user cannot move to are fixed data fields and are subdivided into a static field type which is read from a forms definition file and cannot be modified by any application module, and a write-only type to which the application module can write.

The cursor addressable fields are similarly broken down into read-only and writeable types. The user can only modify the contents of writeable fields to which the cursor can be moved. An error message is produced if the user attempts to edit a read only field. Read only fields are often used to select options from a form which allows forms to serve as menus.

There are also pop-up windows which are often used for sub-menus under a line menu item or to display context sensitive help messages. These windows can be layered to define more pop-up menus or to provide selection of a number of choices (as in selection of a working file from a list of

working files). These windows will disappear when their operation is completed.

Cursor Motion

Motion of the cursor among fields or menu items is accomplished with the TAB and BACKSPACE keys. The TAB key moves to the next field in a line menu, pop-up menu or form window. In the case of forms, only cursor addressable fields are selected and processing occurs in a left to right manner. If necessary the display in the forms window will be scrolled horizontally or vertically to accommodate the next field. The BACKSPACE key similarly moves the cursor to the previous field. Other keys, such as the arrow keys on VT100 terminals will also serve to position the cursor. In pop-up menus, the next and previous fields are defined vertically rather than horizontally.

Line menus and pop-up menus also allow selections by typing the first letter of the desired menu item. If more than one item exists with the same letter, the cursor is moved to the next field with that beginning initial. The initial letter search is not case sensitive.

Forms windows also have implemented additional horizontal and vertical motion keys to make it easier to move within a large form. These keys are defined in the following table. The '^' preceding the key definitions imply Control characters obtained by holding down the control key (Ctrl) simultaneously with the indicated letter key.

ASCII KEY	VT100 KEY*	Cursor Motion Move:	Mnemonic
TAB	(right arrow)	Next field	forwards
BACK	(left arrow)	Previous field	backwards
^N	(down arrow)	Down to next addressable field (may be more than one line)	Next line
^P	(up arrow)	Up to next addressable field (may be more than one line)	Previous line
^D		Down approx. one screenful	Down screen
^U		Up approx. one screenful	Up screen
^T		Top of form	Top form
^B		Bottom of form	Bottom form

*On VT100 terminals, these keys may be used in addition to the standard ASCII control keys on VAX VMS systems.

Abort, Exit, and Help Keys

All of the various user interface modules define the ESCAPE (ESC) key to mean "abort this command or selection and return". In the case of line menus, control will be returned to a prior line menu if possible. In the case of pop-up menus, the current menu will be removed and control returned to either the line menu, or the previous pop-up menu. In the case of a pop-up text display, the window is removed and control is returned to the current working location. In the case of a forms window, this will signal

the end of a forms operation and return the user to the line menu.

In addition to the ESCAPE key, forms windows have additional means of suspending operations. The '/' key is used to make a selection from the line menus, if that operation makes sense within the context of the form in use. A reminder to exit the form first will be issued in those cases for which the line menu selection does not make sense. This situation typically arises when the new line menu function requires the use of the forms window. If a forms window is entered inadvertently, or the user wishes to abort the form without making any changes, then the '^K' key sequence (control K) is used to 'kill' the form. All three form escape keys will put the user back to the line menu.

At any point, the user may obtain context sensitive help by typing the '?' key for a pop-up text display containing a help message. Typing the TAB key in a help screen will page forward through the message, the BACKSPACE pages backwards, and the ESCAPE key exits the help display and returns the user to his previous position. In those instances when the help message is a single line, the command line will be used to display the message and hitting any key will return the use~ to the working point.

Field Edit Mode

In forms windows, the fields which are user modifiable implement a special mode called the field edit mode. In this mode, keyboard input is put into a temporary text buffer and displayed on the forms window. The BACKSPACE key regains its traditional role of backspacing over the previous character in field edit mode. This mode

is entered as soon as a printable character is typed while the cursor is positioned at a modifiable field. An annunciator on the annunciator line is displayed to indicate entry into this mode.

If field edit mode is entered inadvertently, the user may restore the field contents to their previous value by either backspacing to the beginning of the field (e.g. completely erasing the characters entered into the text buffer), or by hitting the ESCAPE key. The ESCAPE key will exit field edit mode, while backspacing does not. In either case, the contents of the field are redrawn with their previous contents and the text buffer is cleared.

To exit field edit mode, the RETURN key is typed which causes the text buffer to be written into the current field. The field will be redrawn to reflect the new contents. If the field contains numerical or logical data, the new contents of the field will be checked to see if it can be interpreted appropriately. An error message is issued if this fails and the user is left in field edit mode to correct the error. It is therefore not possible to put invalid data into a numerical or logical field. No field checking is possible for character data fields, except in certain context sensitive situations (such as specifying the names of files which must already exist).

A Special Note for VMS Users

Because the VMS screen handler for the VT100 family allows for a number of special, non-ASCII function keys (such as the left, right, up, and down keys), it is necessary to hit the ESCAPE twice whenever it is used. This is only true for VMS systems.

5.3 OBJECT MANIPULATION

The object manipulator is called by choosing the "Object" field in any of the module line menus. It translates ANVIL 5000 IGES output files into Object Definition Files in two steps. First the IGES file is translated to an intermediate file called the Object Building Block file. This file contains a complete object definition, except for surface material definitions. The second step defines the material names and places the material definitions in the final, Object Definition File.

Frequently during analysis, satellite surface materials will be redefined or modified without changing the geometry of the object. Since the intermediate file is still available, the Tools allow the material type or specific material characteristics to be changed without requiring the user to return to the model definition stage. This surface material editing also prevents one from needing to redefine all of the surfaces again just to change one of them.

The object manipulator line menu, in addition to the other module names, has two submodules which handle the steps described above. The line menu entry corresponding to creating an Object Building Block file is "Translate". The "Edit Object" field calls a form menu which is used to edit the surface materials. Upon exiting the form, the material definitions are added to the Object Building Block file to create an Object Definition File, the satellite input file used by the analysis codes.

5.3.1 Translating IGES Files to Object Building Block Files

The "Translate" form is used to convert ANVIL 5000 IGES files (.TAP suffix) into an Object Building Block file (.OBB suffix). This is the first stage of the object definition for use by the analysis code. The final step is to define the surface materials.

After selecting a TAP file, the fields in this form are used to define the problem name (the prefix appended to the data files), as well as a comment. It is also possible to view the contents of the input and output files of translation routine. The problem name can be changed from the default by modifying the "Problem Name:" field. A comment may be added to the run history via the "Problem Description" field.

Upon exiting the form, the IGES file is translated to an intermediate file with labels marking the different surface material names. The input IGES file and the output Object Building Block files are both ASCII files and may be viewed using this submodule, if the YES field is picked. The IGES format TAP is very difficult to read and not usually helpful to the user. The OBB file on the other hand is a complete object definition file with the surface materials defined by the variable names MA01, MA02, etc.

The translator traps errors and passes a flag to this module. Errors are found within the OBB file near lines starting "###." It also checks the satellite definitions to see if they contain building blocks which are not NASCAP/GEO or POLAR 1.1 compatible. The translator counts the number of different surface materials and saves this number for

later use by the "Edit Object" submodule. This module assigns material definitions found in material libraries to the satellite surfaces (please see 5.3.2).

5.3.2 Editing Object Definition Files

The "Edit Object" submodule is used to assign the surface materials to the satellite's surfaces. A file created by the Translate command (.OBB suffix) is converted to an object definition file for input to the desired analysis code.

The fields, "Object Name" and "Object Description," can be used to change the problem prefix and to add any desired comments. The rest of the form is used to define the surface materials.

The leftmost column is the material number used by the analysis code for that material type. The second column is the color ANVIL 5000 used to display the material. NOT USED indicates the particular material number is not defined on the object. (See Section 4.4.3 for more information.)

The last two columns are used to define the material name. Either the abbreviated material name (first four characters) or the full material name can be used. As names are entered, the material libraries in the user's path are searched for materials with the same name or abbreviation. The first occurrence of the material name is used.

After all of the surface materials have been defined and found in a materials library, the final Object Definition File is created which contains a complete object definition with surface material definitions.

Material definitions may be taken from any material library database found in any of the directories in the user's directory search path. A protected default material library database is always present in the search path and contains all of the default material definitions described in the NASCAP Programmer's Reference Manual (Reference 1) and the POLAR User's Manual (Reference 2). The contents of the default material database are shown in Appendix B.

5.3.3 Differences between NASCAP/GEO and POLAR 1.1

NASCAP/GEO and POLAR 1.1 use the same building blocks and keywords to define their analysis objects. There are a few building blocks which are not recognized in both analysis programs. NASCAP/GEO does not recognize the SLANT building block. POLAR 1.1 does not recognize BOOM, ATET, ASLANT, APLATE, or zero width wedges (thin triangles). The two codes also differ in how the object may touch the object grid boundary. NASCAP/GEO objects may touch the edge of the object grid boundary at a point or an edge, but not upon an entire surface (Booms may extend out of the grid). POLAR 1.1 objects may not contact the grid at all. Please see Section 4.1 for a detailed discussion on the differences and restrictions of model definition.

5.4 CHARGING ANALYSIS TOOL (SuChgr)

The charging analysis tool is reached by choosing the "SuChgr" field found in any of the module line menus. This module provides access to the material and environmental databases in the directory search path, material and environmental editors to modify or view the definitions in detail, and a spreadsheet tool which computes equilibrium charging potentials of materials in various environments.

This tool provides an estimate of surface potentials using a one dimensional, orbit limited algorithm. No shadowing or surface to surface effects are taken into account. The results from the tool can provide a good starting point for a three dimension calculation by the analysis codes. A good first estimate of the final equilibrium surface potentials can greatly reduce the time needed for the analysis codes to converge on a solution by reducing the number of interations required to find surface potentials in the three dimensional problem.

The material and environment editors and library also provide a convenient way to define and save frequently used definitions. Each directory may have an environment or material database, so any number of local definitions can be saved.

In addition to the other modules, the main line menu for the material charging module contains the following fields to call the submodules: "MatEd" to call the material editor, "MatLib" to call the material library, "EnvEd" to call the environment editor, "EnvLib" for the environment library, and "SSheet" to reach the material charging spreadsheet.

5.4.1 Using the Material/Environment Analysis Spreadsheet

The "SSheet" form is the surface material/environment spreadsheet. It consists of some special functions, environment and material names, and the equilibrium potentials for each material and environment combination. The "Needs Calc/Calc Done" field indicates whether the potentials shown are up to date with the specific materials and environments.

The special functions are:

Setup SUCHGR - calls a form to modify the parameters used to calculate the equilibrium potentials.

Calc Voltages - causes the equilibrium potentials to be calculated. This is necessary if Needs Calc appears in the upper right corner of the form.

Junk an Item - chosen if a material or environment is to be removed from the spreadsheet.

Material Fields:

If a printed material name is picked, the Material editor is entered for that material. If a blank material field is chosen, the Material Library is called.

Environment Fields:

If a printed environment name is picked, the Environment Editor is entered for that environment. If a blank environment field is chosen, the Environment Library is called.

Potential Fields:

If a calculated potential is picked, the initial and final surface and conductor potentials and current sources are displayed.

The following SuChgr parameters may be set from this form:

Initial Material Potential	- starting surface voltage.
Initial Conductor Potential	- starting conductor voltage.
Solar Intensity	- the fractional intensity of the solar radiation.
Fractional Error in Potential	- the amount of error allowed final potential. If the iterative potential solution is within this range of the final answer, it quits.
"Zero" Current	- the amount of current allowed at the final iteration.

The equation which is solved to find the equilibrium potential is

$$J_{tot} = J_i\ (1 + \alpha\ Y_i) - J_e\ (1 - \alpha\ Y_e - B_e) \\ + \alpha\ J_{ph} - \sigma\ (V - V_c)$$

where

- J is flux.
- Y is secondary yield.
- B is backscatter.
- V is voltage.
- α is the effect of the electric field trapping low energy electrons.
- σ is conductance.

and the suffixes are tot (total), i (ion), e (electron), ph (photoemission), and c (underlying conductor).

The fluxes are voltage dependent quantities. Voltages are guessed until the absolute value of J_{tot} is less than the "zero" current value, 10^{-10} amps by default, or the voltage is within three percent of the actual answer.

The search algorithm guesses two voltages until the sign of the total current changes. After the crossover is bounded, the average of the two bounding potentials is tested.

The solar intensity is defined to be the amount of the full intensity at the Earth's surface. This quantity should be a value between 0 and 1.

5.4.2 Material Libraries

The purpose of the "MatLib" module is to browse through the material database files in the current directory search path and select specific definitions for use in either the charging analysis or object definition.

Each directory in the directory search path may contain a material database file. The material database file is used as a library for material definitions. Default material definitions are saved in a write protected data file which is always available. New or modified material definitions may be saved in any material database file which is owned by the current user.

The "MatLib" form has two sections, the view control keywords at the top and the actual material names at the bottom of the form. To choose a material, move the cursor

until the material's name is highlighted. To get a new set of material names, use the view control fields at the top of the form.

The view control keywords are:

Delete Name	-	Allows the user to delete materials from the material library in the names directory.
Backup List	-	Displays the previous block of material names in the current data file.
More Names	-	Displays the next block of material names found in the current data file.
New File	-	Displays the current directory search path so that a new material data file can be chosen.

5.4.3 Editing Materials

The material editor is used to view or modify material properties. The material properties can be modified by moving to the appropriate field and entering the new material value.

Unless this material property is saved from this form, the material will disappear at the end of this session. The first two fields allow you to modify the material data base files which belong to you. The commands are:

Get Material Def	-	Use the Material Library to get a new material definition. (The current material will be lost.)

Save Material Def - Save this material in the named directory's material data file. This will replace an existing material which has the same material name. This function can only be performed on files which you are allowed to modify.

5.4.4 Environment Libraries

The purpose of the "EnvLib" module is to browse through the environment database files in the current directory search path and select specific definitions for either charging analysis or inclusion in a rundeck.

Each directory in the directory search path may contain an environment database file. The environment database file is used as a library for environment definitions. Default environment definitions are saved in a write protected data file which is always available. New or modified environment definitions may be saved in any environment database file which is owned by the current user.

The Environment Library Catalog selects an environment definition for editing or for performing environment/surface interaction analysis. The form has two sections, the view control keywords at the top and the actual environment names at the bottom of the form. To choose an environment, move the cursor until the environment's name is highlighted. To get a new set of environment names, use the view control fields at the top of the form.

The view control keywords are:

Delete Name	-	Allows the user to delete environments from the environment library in the named directory.
Backup List	-	Displays the previous block of environment names in the current data file.
More Names	-	Displays the next block of environment names found in the current data file.
New File	-	Displays the current directory search path so that a new environment data file can be chosen.

5.4.5 Editing Environments

The Environment Editor is used to modify environment properties. The environment properties can be modified by moving to the appropriate field and entering a new value.

Unless this environment property is saved from this form, the environment will disappear at the end of this session. The first two fields allow you to modify the environment data base files which belong to you. The commands are:

Get Environment Def	-	Use the Environment Library to get a new environment definition. (The current material will be lost.)

Save Environment Def	-	Save this file in the named directory's environment data file. This will replace an existing environment which has the same environment name. This function can only be performed on files which you are allowed to modify.

The "Save" field can also be used to write the environment into a temporary file in the keyword format appropriate to the analysis code associated with the control program. The temporary file is usually the flux file, FORTRAN data file 22, for NASCAP/GEO. When using POLAR 1.1, any file name may be used. Later, the rundeck editor can read the environment definition into the NTERAK or SHONTL input file.

The actual environment components NASCAP/GEO and POLAR 1.1 are separated into two sections in the Editor form. Values entered into fields which have equivalent fields in the other analysis code will update those fields automatically. For example, if the low energy Maxwellian density in POLAR 1.1 section is set at $10^{11}/m^3$, the proton and electron densities in first Maxwellian fields of the NASCAP/GEO section are also set.

Only the pertinent environment components which are used to calculate equilibrium potentials are written to the keyword format save files.

5.5 JOB CONTROLLER TOOL (NASCAP OR POLAR)

This module is called by choosing the "NASCAP" (for GEOCAT) or "POLAR" (for POLCAT) fields from one of the module line menus. The job control tool is the interactive user interface to the batch oriented analysis code. The

submodules are used to create valid rundecks for analysis tool programs, check on the status of batch runs, and start, restart, or stop background analysis runs.

This tool is used when the time has come to model the object using the appropriate analysis code, NASCAP/GEO or POLAR 1.1. Previously, the object has been defined (creating an Object Definition File) and studied using the Charging Analysis Tool. A rundeck editor is provided which checks the rundecks to make sure only recognized keywords have been used, attaches data files to the batch runs, and provides an interactive interface which can be used to start and stop a calculation.

In addition to the other modules, the line menu for the Job Control Tool contains the fields which summon the submodules. The "Editor" field sends the user to the rundeck editor submodule. A form is used to choose an existing rundeck or to define a new one and then enter the editor. To list the current status of the batch runs having data files in the current directory, the "Info" field is chosen. Starting, restarting, or stopping analysis jobs is done in the "Control" submodule.

5.5.1 Overview

The Job Control tools come into play after creating the Object Definition File. SuChgr has possibly been used to define an environment and predict final potentials for each material type. These estimates can reduce the length of a calculation by providing good initial values for the surface potentials.

The first step in starting a new analysis run is to create a rundeck. Usually an existing rundeck from a previous calculation is modified to suit the new run. From the environment editor, one can save the environment in a keyword format to a temporary file. In NASCAP/GEO, this would typically be the flux file, unit 20. POLAR 1.1 expects environment definitions in the runstream. The Rundeck Editor could be used to read in keywords from the temporary file into the NTERAK rundeck. Both NASCAP/GEO and POLAR 1.1 have standard test cases included with the software. These are included in Volume 2 of this user's manual. The rundeck can either be created outside of the Tools package or with the simple line editor within the Tools.

The Rundeck editor in the Tools can read in other text files. This is useful if one only has to make a few changes. When exiting from the Rundeck editor, the keyword input to the analysis code can be checked to make sure only recognized keywords are being used.

After a rundeck has been satisfactorily defined, the Control form can be used to start a batch run. The form assigns the input and output files and checks to see if the proper data files are present. A batch job is then submitted. The "info" field of the Job Controller can be used to check the progress of the calculation.

Both analysis codes provide means of restarting calculations. Sometimes it is desirable to stop a job in such a way that it can be restarted later. For example, scheduled computer downtime or high daytime computer loads

may prevent a calculation from being run to completion. Or perhaps the user wants to see how a long calculation is behaving. The Control form will create a file indicating the analysis run should stop smoothly at the next opportunity. The data files could be copied to another directory, using the Tools or via the operating system. The calculation can then be restarted from where it was halted and postprocessing could be performed at the same time in another directory.

In summary the Job Controller provides a mechanism of starting, checking the status, or smoothly halting analysis code calculations.

5.5.2 Rundeck Editor

The initial form encountered upon entering the Rundeck editor determines what files are used to edit a rundeck. For information on what should be included in the rundeck, consult the NASCAP and POLAR 1.1 user's manuals. The module name should be set if keyword validity checking is desired later. The input rundeck is the file which is read into the editor. The output rundeck is the file the editor writes to upon exiting. The rundeck description is a means of adding a comment. The editor may be used for any text file, not just rundecks, if so desired.

The simple editor provided within the Tools is a line editor. Lines of keyword options may be added, replaced or deleted. Other files may also be read into the editor. Below is a description of the keys used to edit a file in this editor.

SIMPLE EDITOR KEYS

Return	-	no action.
TAB,Control N	-	moves to the next line.
Backspace,Control P	-	moves to the previous line.
Letters,Numbers,...	-	go into field edit mode on this line (see below).
?	-	show a help message.
Control R	-	redraw the display.
ESCAPE,/	-	leave the editor, accepting any changes made.
Control K	-	leave the editor, rejecting any changes made.
^ (caret)	-	insert the contents of a file at this point.
Control O	-	open a new line above the current one.
Control A	-	open a new line below the current one.
Delete	-	begin cut mode;

While in Cut Mode lines to be cut are highlighted and:

Delete	-	adds the current line to the cut buffer.
ESCAPE	-	aborts the cut operation.
Backspace	-	'uncut' the previous line.
Return	-	removes cut lines and saves them in the cut buffer.
Control Y	-	yank the contents of the cut buffer above the current line.

Operations involving the cut buffer will warn you if the cut buffer is not empty and ask you if you wish to add to or delete the current buffer contents.

FIELD EDIT KEYS

Return	-	accept any changes, leave field edit mode.
Backspace	-	erase previous character, if at the first character, restores initial field contents.
Letters, Numbers	-	enter the character into field.
?	-	show a help message.
Control R	-	redraw the display.
ESCAPE	-	abort field editing, restore field to previous contents.

When an editing session is completed, you will be asked if you want to run the keyword checker. The first keyword on each line is checked against a list of valid options for the appropriate module and analysis code.

5.5.3 Run Status Monitoring

This section discusses the "Info" submodule. This submodule reads the information provided by a running or stopped analysis code run to give a continuing indication of the status of a calculation.

When the "Info" field is picked, the current directory is checked for the presence of a file called RUNNING.LOK. A message is then shown indicating whether or not a batch job is running in the directory. Then the tool looks for a STATUS.JCO file. If it is there, it is displayed on the screen in page mode. Otherwise a message is printed stating that there is no status information.

Both analysis codes periodically print status information in the file. Whenever a module starts or stops or a calculation cycle is completed, the date, time and a brief message are added to the STATUS.JCO file. The analysis codes also check for any Job Control files in the problem directory (see Section 5.5.4).

5.5.4 Job Control

The "Control" form is used to control analysis tool runs. The run is defined, then the desired action is picked. Runs need to have a problem name entered. This is used to choose object definitions and data files. All jobs started with this form require input files which exist even if no input is required by the module.

The analysis code module must be selected. The recognized modules for GEOCAT are:

NASCAP	-	to run NASCAP/GEO
CONBYU	-	to generate MOVIE.BYU format potential contours.
BYUPOT	-	to create MOVIE.BYU format surface potential plots.
BYUMAT	-	to create MOVIE.BYU format surface material plots.

For POLAR 1.1, the modules are:

VEHICL	-	to translate Object Definition Files into the analysis code format.
ORIENT	-	to reorient the Object Definition (see the POLAR User's Manual).
NTERAK	-	to analyze the interactions of the spacecraft and its environment.
SHONTL	-	to postprocess the data files and create MOVIE.BYU plots.

The input and output text files need to be defined for the batch runs. A comment may be included for later reference.

Actions which can be taken from the form are:

START	-	start a new batch run in the current directory.
STOP GRACEFULLY	-	the run in the current directory will smoothly exit as soon as possible.
KILL	-	the job will terminate itself the next time it checks for control files. It may be more desirable to kill a job at the system level in most circumstances.

Sometimes a job will not complete normally either due to a system crash, direct termination by the user, or possibly a calculation error. When this occurs, spurious signal files may remain. Any files ending with ".JCI" should be deleted. The RUNNING.LOK file should also be removed. If they are present, a job may quit unexpectedly or may not even start. For example, the error message "There is already a Job running this directory" might be received after a system crash.

5.5.5 Differences Between NASCAP/GEO and POLAR 1.1

The two analysis codes use different techniques to define and analyze objects. NASCAP both defines and calculates the interactions with one executable ("NASCAP"), while POLAR uses separate executables for object definition ("VEHICL") and analysis ("NTERAK"). In this context, object definition refers to the translation of the Object Definition File into an internal data format more suitable for calculation.

Rundeck options for each code differ since they solve problems using algorithms suited to the physics involved in their respective regimes. It should be noted that NASCAP is designed to model the response of a spacecraft as its environment changes. This is why the environment definition is read from the flux file rather than from the input rundeck as it is in NTERAK.

Several independent postprocessing drivers are used to produce plots of the computational results and to look at surface data from NASCAP runs. The SHONTL executable performs all of the postprocessing functions for POLAR.

For further detailed information on how to run NASCAP and POLAR, please see the respective user's documentation.

5.6 POST PROCESSING

This module may be exercised by selecting the "PostProc" line menu field in any of the line menus. It provides a means of viewing the analysis code output and data file contents. Since different physical processes are important in NASCAP/GEO and POLAR 1.1, there are differences in the type of information which is available in the data files. Currently only printed output files can be viewed within the form pager, and the TERMTALK program can be used to view surface data from NASCAP/GEO runs.

This tool is particularly useful during satellite analysis for providing information about specific surfaces or sets of surfaces. It is also useful as a non-graphical means of previewing the calculation results.

Besides the other modules, the line menu for the Post Processing module contains the fields which call the following submodules. The "View Output" submodule will page through a text file. The "Surface Data" field is chosen in order to view surface data using TERMTALK.

Note: This is currently only available for NASCAP files.

5.6.1 Viewing Analysis Code Output Files

When the "View Output" field is picked, the user is prompted for the output file name. If possible, a reasonable guess is made at the file name, and it is offered

as a first choice. This utility may be used to view any ASCII file.

5.6.2 Viewing Surface Data

This tool provides an easy means of choosing subsets of the surface data for study using surface properties such as the direction of the surface normal and material types to restrict the quantity of data. The physical data which is available for the satellite surfaces differ in the two analysis codes since the analysis codes are concerned with different physical regimes. Presently, only the data from NASCAP/GEO may be viewed with this tool. To study POLAR 1.1 surface (and grid) data in a written format, use the PRINT option of SHONTL.

There are two methods for viewing surface data. One may view the complete information for a single surface (pick the "Single Surface" field) or use TERMTALK via the "TermTalk" field. TERMTALK is a standalone NASCAP/GEO module used to study the charging history and other data for individual or groups of surfaces. For more information on the use of TERMTALK, see the NASCAP Programmer's Reference Manual or the TERMTALK and MATCHG Handbook.

5.7 FILE UTILITIES

The file utilities module is reached by choosing the "Files" field on any of the module line menus. It is used to manipulate files and directories while running GEOCAT or POLCAT. Utilities are available which delete, list, or copy files or directories. New directories can be created. The current working directory may also be changed. Individual system commands may be issued from this menu.

This module is used to make backup copies of data files, to set up new problem directories, and to clean up old problem directories. It serves as a convenient method of dealing with the analysis code data files.

All of the other modules may be called from this module's line menu. The following submodules manipulate the files and directories used by the CAE Tools: the "Del" submodule is used to delete files, "Copy" makes duplicates of a file or a set of files, the "MkDir" submodule creates a new subdirectory in the current working directory, "NewDir" allows the user to change the current working directory, and "System" provides a means to issue system commands which are executed immediately.

5.7.1 Deleting Files

After picking the "Del" field, the user is asked for a file name to delete. If a question mark is entered instead of a file name, the directory's contents are shown in a popup menu. The chosen file is then deleted.

Deletion commands are not executed immediately; they are written to a temporary file. When the Files module is exited, the user will be asked if he wants the commands he has selected to be executed. Answering no will prevent any files from being deleted.

5.7.2 Listing Files

To list files, use the "System" command. After the appropriate listing command is executed, the output will be displayed in the pager tool. Any operating system command may be issued at the prompt.

5.7.3 Copying Files

When the "Copy" field is chosen, the contents of the current directory are displayed in a popup menu. After a file is chosen, the user is asked for the name of destination file.

This command is not executed immediately. It is written to a temporary file. When the Files module is exited, the copy commands will be executed, at the user's option.

It should be noted that if a file has already been deleted using the "Del" command (Section 5.7.1), it will not exist when the copy command is issued. This will generate an error on most operating systems. It may be desirable, in some cases, to use the "System" function (Section 5.7.6).

5.7.4 Creating a New Directory

The "MkDir" field will create a new directory from within the Tools. The user is prompted for the name of the new directory, relative to the current directory. This command is executed immediately.

5.7.5 Changing the Current Working Directory

The "NewDir" allows the user to change the current working directory. The user may switch to a directory in the directory search path or another directory. If another directory is desired, pick the last popup entry, "Other." The function takes place immediately.

5.7.6 Issuing System Commands

The "System" field provides a means for issuing system commands which are acted on immediately. The output from the command is viewed with the paging tool.

5.8 THE MISCELLANEOUS FORM

To enter the Miscellaneous form, choose the "Misc" menu item in the line menu of any of the other modules. When ever GEOCAT or POLCAT is executed, certain user variables need to be defined. They are quantities like the user's home directory, the current directory, what level of expertise the user prefers, the directory search path which should be followed when looking for data files, and so on. Typically, the defaults are acceptable or even preferred. But as a person becomes more sophisticated in the use of the Tool programs, they may want to customize their local or global user environments.

The Miscellaneous form serves as an editor of the user environment. It is able to save new environment files or just modify the current execution of GEOCAT or POLCAT. A file can be saved in either the current directory or the user's home directory which defines the environment for both or either of the Tools.

When a GEOCAT or POLCAT run is started, the Tool looks for a user environment definition file. The search order is to first look for the specific Tool's definition file, then a general Tool environment file. The current directory is searched first, then the user's home directory.

If no environment file has been found, the system default is used. At end of a Tool run, a new environment file is created. The old environment file is also saved.

5.9 ADDITIONS AND MODIFICATIONS TO THE NASCAP PROGRAMMER'S MANUAL

There are no additions or modifications necessary to the NASCAP Programmer's Reference Manual.

5.10 ADDITIONS AND MODIFICATIONS TO THE POLAR USER'S MANUAL

The following keywords have been added to SHONTL to produce BYU format graphics files and surface value plots (in BYU format):

BYU (*OFF*, ON)	Flags whether or not graphics files are to be written in BYU format.
SURFPLOT iprop	Produces a BYU format plot of the iprop data. BYU must be ON. iprop must be surface data, see Mrbuf, Section 5.33 of the POLAR User's Manual.

The value surrounded by *'s is the default. The *'s are not part of the keyword. Keywords are in capital letters and variables are in lower case.

6. GRAPHICS OUTPUT

Several forms of graphics output are available from the CAE Tools package to display the model and the analysis results. The satellite model may be viewed using ANVIL 5000 during or after object definition. This is not the most informative method since the object surfaces are drawn in the order of definition, without any hidden line removal. The analysis programs will make three dimensional displays of the model which can be viewed using the MOVIE.BYU or DYNA-MOVIE software. MOVIE.BYU is designed to make spectacular solid-filled color plots for a single photograph or overhead slide. DYNA-MOVIE allows the user to perform dynamic three dimensional rotations of data and can be used to make movies or videos of data.

The analysis calculations produce surface charging, space potentials and densities, particle trajectories, and particle number density information which can be viewed graphically. Other output includes three dimensional displays of the model and current versus voltage plots for different materials in various environments. The analysis codes each have methods of producing plots in a batch mode, either during the calculation or afterwards using an analysis code post processor. Typically, without using any of the CAE Tools package, a machine independent file would be created and the pictures viewed on a terminal or printer using a device specific driver.

A more standard, device independent graphics file is now available which uses a MOVIE.BYU/DYNA-MOVIE format. The advantage of this new format is that the machine interfacing is done by a moderately priced piece of commercial software.

Additionally, all of the features of MOVIE.BYU and DYNA-MOVIE can be used without creating a graphics program for both analysis codes.

The following sections discuss the types of graphics output created by the analysis codes, both in the original format and in the MOVIE.BYU format, and some instruction on the use of some basic MOVIE.BYU commands. For a complete study of the MOVIE.BYU and DYNA-MOVIE capabilities, please see the references provided by their distributors (MOVIE.BYU Training Text).

6.1 NASCAP/GEO AND POLAR 1.1 GRAPHICS OUTPUT

NASCAP/GEO and POLAR 1.1 perform similar calculations and produce similar graphical output. They calculate potentials within the three-dimensional grid and on the satellite surfaces. Grid potentials are displayed by plotting on plane of nodal values perpendicular to one of the mesh axis. These planes of values are two dimensional slices or cross sections of the data. Surface potentials and materials are plotted as three-dimensional objects. The graphics files produced by the Tools are in the output format expected by MOVIE.BYU. Any graphics program which can read these ASCII files can display the data.

NASCAP Output

The two codes have different methods of producing graphics output. NASCAP/GEO has three separate plot drivers which can be called by the Job Controller in GEOCAT. The modules are CONBYU, BYUPOT, and BYUMAT. BYUPOT and BYUMAT, the surface potential and material drivers, respectively, require no input other than the NASCAP/GEO FORTRAN data

files. CONBYU produces two-dimensional potential contours and needs four keywords to define a plot. They are:

NZ i	number of z slices in problem (i is an integer)
NG i	number of grids in problem (i is an integer)
AXIS [x,y,z]	axis perpendicular to slice, either x,y, or z
SLICE i	location on perpendicular axis of the slice (i is an integer)

For example, the following would create a contour plot of the potentials in the inner two grids of a calculation in the Y=3 plane:

```
NZ 33
NG 2
AXIS Y
SLICE 3
```

POLAR 1.1 Output

The SHONTL module of POLAR 1.1 handles all of the graphical postprocessing needs of POLCAT. Two new keywords have been added to SHONTL to enable the user to make BYU format plots. They are

BYU ON	causes graphics to be saved in BYU format.
OFF (default)	
SURFPLOT data-type	Makes a surface plot of "data-type"

The BYU keyword is used to put SHONTL into a BYU output mode. The SURFPLOT keyword is used to generate surface plots of "data-type," where "data-type" is any surface information saved in the POLAR 1.1 database. The following are some of the valid "data-types":

Data-type	Description
SRFV	surface voltages
IMAT	material numbers
ICND	underlying conductor numbers
FTOT	total current to a surface

Please see the POLAR 1.1 User's Manual for a complete list.

The following sample rundecks make the different plots.

Example 1

```
COMMENT POTENTIAL CONTOUR
PLOTS ON
BYU ON
POT X
EXIT
```

Example 2

```
COMMENT SURFACE POTENTIAL
PLOTS ON
BYU ON
SURFPLOT SRFV
EXIT
```

Example 3

```
COMMENT SURFACE MATERIAL
PLOTS ON
BYU ON
SURFPLOT IMAT
EXIT
```

It should be noted that only one plot should be made per run to prevent overwriting earlier pictures.

6.2 USING MOVIE.BYU

The general elements of the MOVIE system are Fortran programs to display and manipulate data representing mathematical models whose geometry may be described in terms of polygonal elements, polyhedral solid elements, or contour line definitions. The principal module used in conjunction with the CAE Tools is the DISPLAY program which uses a postprocessor on a high resolution color terminal. DISPLAY provides many graphic options such as: Z-buffering, hidden line removal, outline of surfaces, and shading. DISPLAY has it's own interactive graphics command language which is described in great detail in the MOVIE.BYU Training Text (see your system manager to obtain a copy). This section briefly describes the basics of using DISPLAY with the CAE Tools and provides some examples.

6.2.1 Data Format

Two files are generated by the CAE Tools for use in the DISPLAY module; a geometry file and a surface data file. The geomet. file describes the object to DISPLAY and the surface data file contains an independent quantity (such as surface potentials) for each node in the object.

The geometry data file generated by the CAE Tools for use in DISPLAY is generated using the following Fortran code fragment:

```
       WRITE(6,10) NP,NJ,NE,NEDGE,IFMT,LCE,NEW
10     FORMAT(5I5,/2I5)
       WRITE(6,20) ((VERTX(I,J),I=1,3),J=1,NJ)
20     FORMAT(1P6E12.5)
       WRITE(6,30) (ILMNTS(I,J),I=1,NV,J=1,NE)
30     FORMAT(16I5)
```

where the variables names are defined as:

NP	Number of parts (always 1 in our case)
NJ	Number of vertices or nodes
NE	Number of elements or surfaces
NEDGE	Same as NJ if no double points
IFMT	Always 0, implies the use of MOVIE.BYU
LCE	Length of connectivity card (1)
NEW	Number of nodes in intermediate file (=NPT)
VERTX	x,y, and z Coordinates for each of the NJ nodes
NV	Number of vertices per element (either 3 or 4)
ILMNTS	Element Connectivity, Counterclockwise numbering of nodes making up the NPT elements

It is assumed that objects generated by the CAE Tools consist of either 3-vertex or 4-vertex polygons. The element connectivity table is ordered in counterclockwise order with last vertex negated to indicate the end of the element. For instance we might have:

1 2 -3

or,

1 2 4 -3

The following is a DISPLAY geometry file for a rectangular box made up of six square boxes. All the surface elements are 4-vertex polygons. Only the 22 external surfaces are defined instead of the complete 29 surface elements:

```
   1   24   22   88    0
   1   22
0.00000E+00 0.00000E+00 1.00000E-01 1.00000E-01 1.00000E-01 1.00000E-01
1.00000E-01 0.00000E+00 1.00000E-01 0.00000E+00 1.00000E-01 1.00000E-01
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 1.00000E-01 0.00000E+00
1.00000E-01 0.00000E+00 0.00000E+00 1.00000E-01 1.00000E-01 0.00000E+00
1.00000E-01 2.00000E-01 1.00000E-01 0.00000E+00 2.00000E-01 1.00000E-01
0.00000E+00 2.00000E-01 0.00000E+00 1.00000E-01 2.00000E-01 0.00000E+00
2.00000E-01 1.00000E-01 1.00000E-01 2.00000E-01 0.00000E+00 1.00000E-01
2.00000E-01 0.00000E+00 0.00000E+00 2.00000E-01 1.00000E-01 0.00000E+00
2.00000E-01 2.00000E-01 1.00000E-01 2.00000E-01 2.00000E-01 0.00000E+00
3.00000E-01 1.00000E-01 1.00000E-01 3.00000E-01 0.00000E+00 1.00000E-01
3.00000E-01 0.00000E-00 0.00000E+00 3.00000E-01 1.00000E-01 0.00000E+00
3.00000E-01 2.00000E-01 1.00000E-01 3.00000E-01 2.00000E-01 0.00000E+00
   4    1    3   -2    7    5    6   -8    1    5    7   -3    6    5    1   -4
  10    4    2   -9   12    8    6  -11   12   11   10   -9   11    6    4  -10
  13    2    3  -14   15    7    8  -16   14    3    7  -15    9    2   13  -17
  16    8   12  -18   18   12    9  -17   13   14   20  -19   22   21   15  -16
  14   15   21  -20   20   21   22  -19   17   13   19  -23   24   22   16  -18
  23   24   18  -17   23   19   22  -24   17   13   19  -23   24   22   16  -18
```

A second data file required by DISPLAY is the surface data file which contains any independent value for each node in the geometry file. The CAE Tools can generate two types of surface data files; a material file which identifies the material for each surface, and a surface potential file. Each vertex is assigned a value. The values are written using the following Fortran code fragment:

```
DO 70 I = 1,NJ,6
    WRITE(6,130) (VALUES(J),J=I,I+5) 130
    FORMAT(1P6E12.5)/
70  CONTINUE
```

For example, a potential file for the object described previously would look as follows:

```
-9.00000E+00-9.00000E+00-8.90000E+00-8.90000E+00-8.90000E+00-8.90000E+00
-8.90000E+00-9.00000E+00-8.90000E+00-9.00000E+00-8.90000E+00-8.90000E+00
-9.00000E+00-9.00000E+00-9.00000E+00-9.00000E+00-8.90000E+00-9.00000E+00
-8.90000E+00-9.00000E+00-9.00000E+00-8.90000E+00-8.90000E+00-9.00000E+00
-8.90000E+00-8.80000E+00-8.90000E+00-9.00000E+00-8.80000E+00-8.90000E+00
-9.00000E+00-8.80000E+00-9.00000E+00-8.90000E+00-8.80000E+00-9.00000E+00
-8.80000E+00-8.90000E+00-8.90000E+00-8.80000E+00-9.00000E+00-8.90000E+00
-8.90000E+00-9.00000E+00-9.00000E+00-8.80000E+00-8.90000E+00-9.00000E+00
-8.80000E+00-8.80000E+00-8.90000E+00-8.80000E+00-8.80000E+00-9.00000E+00
-8.70000E+00-8.90000E+00-8.90000E+00-8.70000E+00-9.00000E+00-8.90000E+00
-8.70000E+00-9.00000E+00-9.00000E+00-8.70000E+00-8.90000E+00-9.00000E+00
-8.70000E+00-8.80000E+00-8.90000E+00-8.70000E+00-8.80000E+00-9.00000E+00
```

6.2.2 Using DISPLAY

The first three prompts always encountered upon first executing DISPLAY are for the data file names for the files previously described:

<READ GEOM FILE>	Name of geometry file
<READ DISP FILE>	Not used by CAE Tools (hit carriage return)
<READ FUNC FILE>	Nodal values file

Both GEOCAT and POLCAT will output geometry files on Fortran logical unit 8. On VMS systems the default file name for unit 8 is FOR008.DAT and this would generally be the response for the <READ GEOM FILE> prompt.

GEOCAT uses Fortran logical unit 9 (FOR009.DAT) to write out nodal values, while POLCAT uses logical unit 7 (FOR007.DAT). DISPLAY may be given any valid file name if the user has previously renamed the default files.

After each prompt, DISPLAY will reply with a message echoing what was read. In response to the geometry file prompt, the number of parts (always 1 in the CAE Tools), the

number of nodes, and the number of elements read are displayed. The <READ DISP FILE> prompt should always respond with zero displacements read since displacement files are not generated by the CAE Tools. The <READ FUNC FILE> will display the number of nodal values read and should agree with the number of nodes read in from the GEOMETRY file. A common error is to mistype file names which results in zero items being input. It is a good practice to know beforehand the number of elements and nodes in the geometry file so that errors can be more easily recognized.

Following the prompts for input file names, DISPLAY is now ready for interactive use. DISPLAY now prompts the user with a '>' character. If more information is required for a given command the prompt becomes a '>>' and if even more information is required, then a '>>>' is issued. New commands should only be issued at the '>' prompt. All commands have default values associated with them which are used when a carriage return is typed in response to a prompt. The defaults are usually reasonable, so if in doubt, typing a carriage return to the '>>' and '>>>' prompts will often suffice.

All input is done with a free format. Entering a sequence of numbers to a command requires only that each value be separated by commas. It should also be noted that any of the DISPLAY commands may be abbreviated by their first four characters.

The next section briefly describes the most basic DISPLAY commands. The example to follow is intended to show only the basic use of the DISPLAY module. Many more features are available in DISPLAY than are touched on here. For more

information, consult the the DISPLAY User's Manual in Appendix A of the MOVIE.BYU Training Text.

6.2.2.1 Frequently Used DISPLAY Commands

The following is a summary of the commands most often used to create graphic output with DISPLAY from CAE Tools output files. These are only a subset of the commands available, and some of the commands have many options. It is worthwhile to note that three fundamental types of display are possible with the DISPLAY module: wire frame (including hidden line removal), surface rendition (with light source highlighting and/or shadowing), and surface value rendition (with contour lines and/or colors). Certain options apply only to specific drawing modes.

COLOR

The COLOR command determines the colors to be used for the background and the object in the foreground. Colors are specified by three numbers representing scales in red, blue and green. The scales are specified as values between 0 (off) and 1 (full intensity). The color red would therefore be specified as 1,0,0, while a light purple would be 0.5,0.5,0. The COLOR command will prompt for both a background color and a color for each part in the object. The CAE Tools always generate only one part as far as DISPLAY is concerned.

DRAW

This command will draw the object on the screen without hidden line removal. It is a quick method for

determining the results of rotations, translations, and scaling commands.

VIEW

The VIEW command displays the object in accordance with the settings set by previous commands. If the object is to be drawn as a wire frame drawing (the initial state), then a hidden line drawing is created. If a surface element drawing is called for (see the SCOPE command), then a drawing is rendered using the settings established with the COLOR and LIGHT commands.

LIGHT

The LIGHT command allows the user to specify one or more light sources for illuminating the object. DISPLAY will prompt for light source vector (as a three number vector from the object origin), the light source intensity (as a number between 0 and 1) and for two parameters related to the reflectivity of the object surface. The light command is required for displaying shadows and highlights on the object surfaces.

ROTATE

This command allows the user to rotate the model about an axis. Rotation of the model is specified as angles of rotation in degrees about the three axes.

TRANSLATE

This command moves the model from one position to another.

SCALE

This command zooms the object to fill more or less of the screen.

SCOPE

This command establishes whether a wire frame or surface shaded object is to be drawn. It also determines whether a coordinate triad is produced and whether any existing drawing is to be erased before the next VIEW or DRAW command is executed. By first drawing a surface shaded plot, then changing the scope to wire frame and not erasing the existing drawing, it is possible to overdraw the object's wire frame over the object surface. This produces a nice effect by outlining all the surface elements.

FRINGE

This command allows the user to establish a color scale to be applied to surface shaded image (such as surface potentials). The user is asked to supply the range over which the color scale applies. It is up to the user to know beforehand what the range of values are in the surface function file and to establish the color scale accordingly. The range of data found in the function file is displayed when the file is first read in.

CONTOURS

This command allows the user to select the number of contours and their range. The scalar functions will then appear as contours on the visible surfaces of the model when the model is displayed using the VIEW command. The user has to remember to set many options before he uses this option.

DIFFUSE

This command allows the user to specify the minimum light intensity for continuous tone images. Do not forget that the CAE Tools make use of only one MOVIE.BYU part.

6.2.2.2 Step by Step Example:

The following is an annotated example of using the DISPLAY module. Input typed by the user is displayed in lower case, although input to DISPLAY is case insensitive.

HOST PROMPT: run display

DISPLAY is now executing, prompting for names for the geometry file and the surface data file. Displacement files are not generated by the CAE Tools. Note the responses from DISPLAY indicating what was obtained from each file.

```
<MOVIE SYSTEM DISPLAY>
<READ GEOM FILE>     for008.dat
<READ:     1 PARTS;    24 COORDINATES;    22 ELEMENTS.>
 <READ DISP FILE>       <cr>
 <READ FUNC FILE>     for009.dat
<PREVIOUS RANGE:> <      .000 <X<       .300      .000 <Y<       .200      .000 <Z<       .100>
<-0.00000E+00 <SCALAR FUNCTION< -0.00000E+00>
<ORIGIN MOVED TO:  .150     .100     .050>
<DISTANCE TO ORIGIN:  1.05 ,ANGLE: 28.00 ,ZMIN:  .10 ,ZMAX:  2.10>
<1 PARTS WITH ELEMENT LIMITS:>
   1  22 >>
```

Use the LIGHT command to define a single light source positioned at the eye of the observer. Use default values for light exponents and surface highlights.

```
> light
 <ENTER LIGHT SOURCE NUMBER (1- 4)>        1
 <LIGHT SOURCE AT EYE OF OBSERVER?>        y
 <LIGHT SOURCE INTENSITY>                  1.
 <PARTS I1/I2, REGULAR LIGHT EXPONENT>   <cr>
>>>
 <HIGHLIGHTS FOR PARTS I1/I2,INTENSITY,EXPONENT>   <cr>
>>>
>>
```

Use the COLOR command to select a red background and ignore the foreground color (This example will produce a surface potential plot which does not require a foreground color). Use the defaults for color fringe selection.

```
> color
<BACKGROUND RED, BLUE, GREEN>    1,0,0
<PARTS I1/I2, RED, BLUE, GREEN>  <cr>
>>>
<STANDARD FRINGE COLORS?>        <cr>
<FRINGE NUMBER, RED, BLUE, GREEN>   <cr>
>>>
```

Use the SCOPE command to select surface rendition mode and to have DISPLAY erase the display prior to every DRAW and VIEW command. In addition, display a coordinate triad.

```
> scope
<1-Line drawing, 2-Shaded Image>        2
<Enter mode>
 <Draw over existing image?>            n
 <Suppress coordinate triad?>          <cr>
>>
```

Use the ROTATE command three times to rotate the object about each axis.

```
> rotate
>>
 <AXIS, ANGLE>        x,30
>>
> rotate
>>
 <AXIS, ANGLE>        y,30
>>
> rotate
>>
 <AXIS, ANGLE>        z,10
>>
```

Use the FRINGE command to set the number of colors to be used for the surface potential contours. Specify the surface potential range of values over which to map color contours and to specify that a fringe bar showing an index for each of the colors is to be displayed (default).

```
> fringe
>>
 <NUMBER OF FRINGES>              5
 <DISPLACEMENT FRINGES?>         <cr>
 <PARTS I1/I2, RANGE X1,X2>    1,1,-9.,-8.9
>>>
 <PARTS I1/I2 WITHOUT FRINGES>   <cr>
>>>
 <SUPPRESS FRINGE BAR?>          <cr>
 <COLOR FRINGE BAR TO REPRESENT FRINGE "GROUP" NO.>
<cr>
 >>
```

Use the VIEW command to obtain a surface potential image on the display device. Surface potentials will be shown using colors to distinguish potentials.

```
> view
```

Re-invoke the SCOPE command to superimpose on the surface potential plot, a wire frame rendition of the object.

```
> scope
 <1-Line drawing, 2-Shaded Image>      1
<Enter mode>
 <Draw over existing image?>       y
 <Suppress coordinate triad?>       <CR> >>
```

Use the VIEW command to display the wire frame on top of the existing display.

```
> view
```

Exit the DISPLAY module

```
> exit
```

6.2.2.3 Contour Plots

Generating the surface potential input files from the two CAE Tools differ slightly. In POLCAT, the FOR008.DAT and FOR007.DAT files are created with the SHONTL module. In GEOCAT, it is necessary to run the CONBYU post-processor in order to generate the FOR008.DAT and FOR009.DAT files. CONBYU is run in exactly the same manner as the CONTOUR command in DISPLAY.

6.2.3 Note to System Managers

The MOVIE.BYU program must be redimensioned and recompiled to allow more than the default number of surfaces. Typically, geometry files are limited to 1000 nodes and 100 surface elements. Many of the POLCAT and GEOCAT models will exceed these limits. We have set the following limits in our version of DISPLAY:

NJMAX=6000	maximum number of nodes
NPTMAX=6000	maximum number of elements
ICNMAX=24000	connectivity array

6.3 OTHER GRAPHICS OUTPUT

Both NASCAP/GEO and POLAR 1.1 have retained their original graphical output systems. If a form of output is desired in the old style from the analysis codes, please refer to the appropriate user's manual. Techniques for alternate forms of graphics are explained in these references.

7. REFERENCES

1. Mandell, M. J., P. R. Stannard, and I. Katz, NASCAP Programmer's Reference Manual, S-CUBED Report SSS-84-6638, 1984.

2. Lilley, Jr., J. R., D. L. Cooke, G. A. Jongeward, and I. Katz, POLAR User's Manual, Report AFGL-TR-85-0246 ADA173758, S-CUBED Report SSS-R-86-7563, 1985.

3. Purvis, C. K., H. B. Garrett, A. C. Whittlesey, and N. J. Stevens, "Design Guidelines for Assessing and Controlling Spacecraft Charging Effects," NASA Technical Paper 2361, 1984.

4. Stevens, N. J., "Design Practices for Controlling Spacecraft Charging Interactions," NASA Technical Memorandum 82781, 1982.

5. Introduction to VAX/VMS. (This is the introductory manual which describes the file system used in VMS.)

6. ANVIL 5000 GRAPL Reference Manual, Manufacturing and Consulting Services, 9500 Toledo Way, Irvine, CA 92718, 1984.

7. Christiansen, H., and M. Stephenson, MOVIE.BYU Training Text, Graphics Utah Style, 1980 North 1450 East, Provo, Utah 84604, 1986.

8. Stannard, P.R., I. Katz and M. J. Mandell, "Additional Extensions to the NASCAP Computer Code, Volume II," NASA-CR-167856, Chapter 3, 1982.

9. TERMTALK and MATCHG Handbook, S-CUBED Manual, January 1985.

APPENDIX A
CAE TOOLS INSTALLATION INSTRUCTIONS

APPENDIX A - CAE TOOLS INSTALLATION INSTRUCTIONS

This appendix contains instructions and information about the system support and installation requirements of the CAE Tools software package and analysis tools.

A.1 VAX/VMS SYSTEMS

The CAE Tools are provided on a VAX/VMS BACKUP tape in a single file called CAETOOLS.BCK. Approximately 60,000 blocks of data are stored on the tape. It is recommended that 100,000 blocks of diskspace be reserved to install and test the CAE Tools.

The analysis codes NASCAP and POLAR have been modified to run on the VMS operating system version 2 and higher. The CAE Tools require the Screen Management Guidelines libraries which became available on VMS systems after version 4.3.

The installer is assumed to have a good working knowledge of the VMS operating system and is an experienced user of the BACKUP utility. The CAE Tools make use of VMS symbols in order to locate files containing form definitions, help messages and default material and environment libraries.

Command procedures are provided to re-compile and re-link all parts of the CAE Tools should that become necessary. The following naming convention has been used:

C<module_name>.COM - Compile and module or library.
L<module_name>.COM - Link a program module.

These command procedures will need to be modified to reflect the appropriate directory into which the CAE Tools have been installed. S-Cubed versions of these command procedures will have 'S3' prepended to each of these procedures. It should not be necessary to re-compile or re-link the CAE Tools on VAX/VMS systems.

A.2 REQUIRED SUPPORTING HARDWARE AND SOFTWARE

The CAE Tools consist of five major software modules:

* ANVIL 5000
* NASCAP/GEO
* POLAR 1.1
* CAE Tools User Interface
* MOVIE.BYU

ANVIL 5000 and MOVIE.BYU are not supplied on the distribution tape and are assumed to be supported at the host site. The distribution tape contains ANVIL 5000 GRAPL routines required to define NASCAP/POLAR building blocks and a graphics tablet overlay description.

The distribution tape for the CAE Tools provides everything required to compile, link, and run NASCAP/GEO and POLAR 1.1.

The CAE Tools User Interface must be linked with the Screen Management Guidelines Library provided on VMS Systems after Version 4.3. Everything else required to compile the CAE Tools User Interface is provided on the distribution tape.

Required hardware for the CAE Tools:

Graphics terminal for use with ANVIL 5000. Tektronix 4207 or equivalent is recommended along with a graphics tablet for use with the graphics tablet overlay.

VT-100 alphanumeric terminals or equivalent for use with the CAE Tools User Interface. Any terminal which has direct cursor addressing capabilities and is described in a VMS Terminal Capabilities File, may also be used.

Graphics display device for use with the MOVIE-BYU program. Typically a color bitmapped display, but any device supported by MOVIE.BYU or DNYA-MOVIE may be used.

A.3 GLOBAL VARIABLES AND FILE STRUCTURE

Several logical symbols are required in the user's LOGIN.COM file in order to successfully run the CAE Tools User Interface. These symbols are required for submitting batch runs from within the Tools and to allow the Tools to locate files containing help messages and form definition files. A command procedure has been provided in the [XXXX.CAETS] directory (substitute the directory name into which the Tools were loaded for XXXX on all directory name references) called DEFINE.COM to create these symbols. The following line must be inserted into the LOGIN.COM file PRIOR to any command which might cause the login procedure to be circumvented in batch mode:

```
$@[XXXX.CAETS]DEFINE.COM
```

Typical batch mode circumvention commands look like:

```
$ IF F$MODE() .EQS. "BATCH" THEN EXIT
```

An alternative method is to have the system manager define the symbols in [XXXX.CAETS]DEFINE.COM as global system logicals.

The directory structure which is used for the CAE Tools is outlined as follows:

```
[XXXX] - Installation directory
   [.ANVDEMO] - Sample IGES files created from ANVIL 5000
   [.ANVIL] - GRAPL programs for the ANVIL 5000
   [.ANVNAS] - GRAPL programs for the ANVIL 5000

   [.BIN] - Executables for all programs in the CAE Tools

   [.CAETS] - CAE Tools User Interface Directory
      [.BYU] - MOVIE.BYU Postprocessors for GEOCAT
         [.BYUMAT] - Material plots program
         [.BYUPOT] - Surface potential plot program
         [.CONBYU] - Contour plot program
      [.GENMOD] - Routines modified from the GENLIB library
      [.INC] - Include files for all of GEOCAT and POLCAT
               routines
      [.IOLIB] - Routines modified from the IOLIB library
      [.OLDCODE] - General purpose utilities
      [.PROBLEMS] - Sample problems
         [.CUBE] - A simple cube for POLCAT
         [.FLTSATCOM] - FLTSATCOM satellite for GEOCAT
         [.MSHUTL] - Micro shuttle for POLCAT
      [.SCREEN] - Screen handler routines
      [.SUCHGR] - Surface Charger routines
      [.SYS] - GEOCAT/POLCAT auxiliary files:
               [XXXX.CAETS]DEFINE.COM defines symbols to
               this directory structure
```

[.FORM] - Form definition files required to display forms

[.GLOB] - Global files: Material and Environment libraries, Default environment files

[.MESS] - Message files for help windows

[.SYSDEP] - System dependent routines

[.SYSTEM] - Main routines for GEOCAT and POLCAT

[.TERMTALK] - CAE Tools version of TERMTALK (for GEOCAT)

[.TOOLS] - User interface routines common to POLCAT and GEOCAT

[.TRANS] - Translator routines for conversion of IGES files

[.NASCAP] - NASCAP/GEO directory tree

[.CONTOURS] - Contour plotting program

[.IOLIB] - IOLIB library source code

[.MATCHG] - MATCHG program

[.NASCAP] - Main routines for NASCAP/GEO

[.OBJDISP] - OBJDISP program

[.PLOTREAD] - PLOTREAD program

[.POTCOLOR] - POTCOLOR program

[.TERMTALK] - TERMTALK program, stand-alone version

[.TESTCASES] - NASCAP/GEO test problems

[.POLAR] - POLAR 1.1 program directory

[.DEMORUNS] - Test problems for POLAR 1.1

[.GENLIB] - GENLIB library source code

[.MSIOTEST] - MSIOTEST program

[.NTERAK] - NTERAK module for POLAR

[.ORIENT] - ORIENT module for POLAR

[.POLLIB] - POLLIB library source code

[.SHONTL] - SHONTL module for POLAR

[.VAXLIB] - VAXLLIB library source code

```
[.VEHICL] - VEHICL module for POLAR
[.VELLIB] - VELLIB library source code
```

A.4 INSTALLING CAE TOOLS SOFTWARE

Make the directory into which the tools are to be installed, set default to that directory and unlcad the backup tape:

```
$ CREATE/DIR [XXXX]
$ SET DEFAULT [XXXX]
$!
$! MTAO: is any tape drive on the host system
$!
$ ALLOCATE MTAO:
$ MOUNT/FOREIGN MTAO:
$ BACKUP/LOG/REWIND/SELECT=[LILLEY...]*.*
  MTAO:CAETOOLS.BCK [...]

    Lots of messages listing files restored from tape
         :
         .
$ UNMOUNT MTAO:
$ DEALLOCATE MTAO:
```

Check to make sure there are files in the directories. The executable files for the CAE Tools: POLCAT and GEOCAT are stored in the directory [XXXX.BIN].

Edit the user's login.com file to include the line:

```
$@[XXXX.CAETS]DEFINE
```

and make sure this is inserted prior to any exits for batch mode such as the following:

```
$IF F$MODE().EQS."BATCH" THEN EXIT
```

After modifying the user's LOGIN.COM, run the LOGIN.COM command procedure to define the symbols for the current working session and to verify that the changes made are correct.

This completes the installation procedure. Create a working subdirectory, copy a sample working problem from the [XXXX.CAETS.PROBLEMS] directory and make a test run using [XXXX.BIN]GEOCAT or [XXXX.BIN]POLCAT. If the CAE Tools fail to locate form definition files or help messages, make sure the DEFINE.COM is correctly implemented in the user's LOGIN.COM file.

A.5 INSTALLING NASCAP/GEO

NASCAP/GEO is automatically installed in the directory [XXXX.NASCAP] from the backup tape unloading described in Section A.4. The executable NASCAP/GEO modules: NASCAP, CONTOURS, MATCHG, OBJDISP, POTCOLOR, TERMTALK, NTEK4014, NTEK4207, PTEK4014, and PTEK4207 are stored in the directory [XXXX.BIN]

A.6 INSTALLING POLAR 1.1

POLAR 1.1 is automatically installed in the directory [XXXX.NASCAP] from the backup tape unloading described in Section A.4. The executable POLAR modules: VEHICL, ORIENT, NTERAK, SHONTL, and MSIOTEST are stored in the [XXXX.BIN] directory.

APPENDIX B

DATA BASES USED BY CAE TOOLS

APPENDIX B - DATA BASES USED BY CAE TOOLS

This appendix contains a listing of the default data bases used by the CAE Tools.

Appendix B - CAE Tools Data Bases

This is the contents of the default material library stored in the file: [XXXX.CAETS.SYS.GLOB]MATLIB.DAT where XXXX is the installation directory name for the CAETOOLS.

```
comment
comment        MATLIB.DAT  (standard system material library)
comment
comment        The material definitions should be sorted alphabetically.
comment        Duplicate names should be avoided, the first definition
comment        is presently used in the case of duplicates.  But this
comment        may change.
comment
comment        The following material definitions are taken directly
comment        from the nascap routine, matdef.
comment
comment        The format of data in this file is:
comment                newmat  MAT_NAME  DATE
comment                matcom  COMMENT
comment                MATERIAL_PROPERTIES(20)
comment        MAT_NAME is the material name (required)
comment                The first four characters must be unique to the file.
comment                These characters (upper case) are used in the analysis
comment                codes as material identifiers.
comment        DATE is the last modification date (optional, but automatic)
comment                format: xx/yy/zz (xx=month, yy=day, zz=year)
comment        COMMENT is a short (80 char) multi word comment (optional)
comment        MATERIAL_PROPERTIES(20) 20 real number values of material
comment                properties.  These numbers may be spread on however
comment                lines it takes, but all 20 values are required.
comment                The material properties and their units are:
comment
comment   VALUE   PROPERTY                     INPUT VALUE UNITS
comment     1     DIELECTRIC CONSTANT              (NONE)
comment     2     THICKNESS                        METERS
comment     3     CONDUCTIVITY                     MHO/M
comment     4     ATOMIC NUMBER                    (NONE)
comment     5     DELTA MAX                        (NONE)
comment     6     E-MAX                            KEV
comment     7     RANGE                            ANG.
comment     8     EXPONENT                         (NONE)
comment     9     RANGE                            ANG.
comment    10     EXPONENT                         (NONE)
comment    11     YIELD FOR 1KEV PROTONS           (NONE)
comment    12     MAX DE/DX FOR PROTONS            KEV
comment    13     PHOTOCURRENT                     A/M**2
comment    14     SURFACE RESISTIVITY              OHMS
comment    15     SPACE DISCHARGE POT'L            VOLTS
comment    16     INTERNAL DISCHARE POT'L          VOLTS
comment    17     RADN INDUCEDCOND'Y COEFF         MHOMS3
comment    18     RADN INDUCEDCOND'Y POWER         (NONE)
comment    19     DENSITY                          KG/M*3
comment    20     undefined                        ---
comment
comment          From matdef (nascap/geo):
comment
comment     MATERIAL PARAMETERS ARE DESCRIBED IN NASA CR-159417, PP.46-47.
comment          See also the Nascap and Polar user documents
comment
newmat    ALUMIN  03/05/81
matcom  This is a NASCAP/GEO default definition.
     1.,.001,-1.,13.,.97,.3,153.7,.8,
     220.,1.76,.244,230.,.00004,-1.,1.E+4,2.E+3,
```

```
       1.E-13,1.,1.E+3,20.
newmat      AQUADG   03/05/81
matcom  This is a NASCAP/GEO default definition.
       1.,.001,-1.,6.,1.,.3,-1.,0.,
       2.,12.,.455,140.,.000021,-1.,1.E+4,2.E+3,
       1.E-13,1.,1.E+3,20.
newmat      CPAINT   03/05/81
matcom  This is a NASCAP/GEO default definition.
       3.5,.001,-1.,5.,2.1,.15,71.48,0.6,
       312.1,1.77,.455,140.,.00002,-1.,1.E+4,2.E+3,
       1.E-13,1.,1.E+3,20.
newmat      GOLD   03/05/81
matcom  This is a NASCAP/GEO default definition.
       1.00,.001,-1.,79.,.88,.8,88.79,.92,
       53.48,1.73,.413,135.,.000029,-1.,1.E+4,2.E+3,
       1.E-13,1.,1.E+3,20.
newmat      INDOX   03/05/81
matcom  This is a NASCAP/GEO default definition.
       1.,.001,-1.,24.4,1.4,.8,-1.,0.,
       7.18,55.5,.49,123.,.000032,-1.,1.E+4,2.E+3,
       1.E-13,1.,1.E+3,20.
newmat      KAPTON   03/05/81
matcom  This is a NASCAP/GEO default definition.
       3.5,.000127,1.E-16,5.,2.1,.15,71.48,.60,
       312.1,1.77,.455,140.,.00002,1.E+16,1.E+4,2.E+3,
       1.E-13,1.,1.E+3,20.
newmat      MAGNES   03/05/81
matcom  This is a NASCAP/GEO default definition.
       1.,.001,-1.,12.,.92,.25,-1.,0.,
       1.74,24.3,.244,230.,.00004,-1.,1.E+4,2.E+3,
       1.E-13,1.,1.E+3,20.
newmat      NPAINT   03/05/81
matcom  This is a NASCAP/GEO default definition.
       3.5,.00005,5.9E-14,5.,2.1,.15,-1.,0.,
       1.05,9.8,.455,140.,.00002,1.E+13,1.E+4,2.E+3,
       1.E-13,1.,1.E+3,20.
newmat      SCREEN    03/05/81
matcom  This is a NASCAP/GEO default definition.
       1.,.001,-1.,1.,0.,1.,10.,1.5,
       0.,1.,0.,1.,0.,-1.,1.E+4,2.E+3,
       1.E-13,1.,1.E+3,20.
newmat      SILVER  05/31/79
matcom  This is a NASCAP/GEO default definition.
       1.00,.001,-1.,47.,1.,.8,84.46,.82,
       79.43,1.74,.49,123.,.000029,-1.,1.E+4,2.E+3,
       1.E-13,1.,1.E+3,20.
newmat      SIO2   03/05/81
matcom  This is a NASCAP/GEO default definition.
       4.,.000127,1.E-14,10.,2.4,.4,116.3,0.81,
       183.1,1.86,.455,140.,.00002,1.E+19,1.E+4,2.E+3,
       1.E-13,1.,1.E+3,20.
newmat      SOLAR  03/05/81
matcom  This is a NASCAP/GEO default definition.
       3.8,.000179,1.E-17,10.,2.05,.41,77.5,.45,
       156.1,1.73,.244,230.,.00002,1.E+19,1.E+4,2.E+3,
       1.E-13,1.,1.E+3,20.
newmat      TEFLON  03/05/81
matcom  This is a NASCAP/GEO default definition.
       2.,.000127,1.E-16,7.,3.,.3,45.37,.40,
       217.6,1.77,.455,140.,.00002,1.E+16,1.E+4,2.E+3,
       1.E-13,1.,1.E+3,20.
```

This is the contents of the default environment library stored in the file: [XXXX.CAETS.SYS.GLOB]ENVLIB.DAT where XXXX is the installation directory name for the CAETOOLS.

```
comment
comment         ENVLIB.DAT  (standard system environment library)
comment
comment         The environment definitions should be sorted alphabetically.
comment         Duplicate names should be avoided, the first definition
comment         is presently used in the case of duplicates.  But this
comment         may change.
comment
comment         The format of data in this file is:
comment                 newenv  ENV_NAME  DATE
comment                 envcom  COMMENT
comment                 ENVIRO_PROPERTIES(20)
comment         ENV_NAME is the environment name (required)
comment                 The first four characters must be unique to the file.
comment                 These characters (upper case) are used in the analysis
comment                 codes as environment identifiers.
comment         DATE is the last modification date (optional, but automatic)
comment                 format: xx/yy/zz (xx=month, yy=day, zz=year)
comment         COMMENT is a short (80 char) multi word comment (optional)
comment         ENVIRO_PROPERTIES(20) 20 real number values of environment
comment                 properties.  These numbers may be spread on however
comment                 lines it takes, but all 20 values are required.
comment                 The environment properties and their units are:
comment
comment        NASCAP/GEO environment properties
comment   VALUE   PROPERTY                       INPUT VALUE UNITS
comment     1     Maxwellian 1, electron temp          eV
comment     2     Maxwellian 1, electron dens          #/m**3
comment     3     Maxwellian 1, ion temp               eV
comment     4     Maxwellian 1, ion dens               #/m**3
comment     5     Maxwellian 2, electron temp          eV
comment     6     Maxwellian 2, electron dens          #/m**3
comment     7     Maxwellian 2, ion temp               eV
comment     8     Maxwellian 2, ion dens               #/m**3
comment
comment        POLAR 1.1 environment properties
comment   VALUE   PROPERTY                       INPUT VALUE UNITS
comment     9     Ion Mass                             AMU
comment    10     Ion to H+ # density ratio            (none)
comment    11     Cold Maxwellian Temperature          eV
comment    12     Cold Maxwellian Density              #/m**3
comment        Energetic Electrons
comment    13     Power Law coeff                      (m**2*sec*str*eV)**-1
comment    14     Power Law exponent                   (none)
comment    15     Power Law low integration limit      eV
comment    16     Power Law high integration limit     eV
comment    17     Maxwellian Temperature               eV
comment    18     Maxwellian Density                   #/m**3
comment    19     Gaussian coeff                       (m**2*sec*str*eV)**-1
comment    20     Gaussian Peak location               eV
comment    21     Gaussian Peak e-fold width           eV
comment
comment          See also the Nascap and Polar user documents
comment
newenv     DMSP  08/01/87
```

```
envcom  This is a POLAR 1.1 DMSP environment (see manual).
        .2, 1.0e+11, .2, 1.0e+11, 4.3e+3, 4.2e+6, 0., 0.
        16., 1.0e20, .2, 1.0e+11, 1.4e+12, 1.2, 50., 1.0e+6
        4.3e+3, 4.2e+6, 8.8e+5, 8.2e+3, 1.8e+3
newenv     POLARDEFAULT  08/01/87
envcom  This is the POLAR 1.1 default environment.
        .2, 1.0e+11, .2, 1.0e+11, 0., 0., 0., 0.
        16., 1.0e20, .2, 1.0e+11, 0., 1.2, 1.0e+02, 1.0e+09
        0., 0., 0., 0., 0.
newenv NASCAPDEFAULT                                 04/10/87
envcom FLTSATCOM default environment for nascap.
  0.1500E+05,  0.1000E+07,  0.0000E+00,  0.0000E+00,  0.0000E+00,  0.0000E+00,
  0.0000E+00,  0.0000E+00,  0.0000E+00,  0.0000E+00,  0.0000E+00,  0.0000E+00,
  0.0000E+00,  0.0000E+00,  0.0000E+00,  0.0000E+00,  0.0000E+00,  0.0000E+00,
  0.0000E+00,  0.0000E+00,  0.0000E+00,
```